

**International Doctorate School in Information and
Communication Technologies**

DIT - University of Trento

**A COLLABORATIVE PLATFORM FOR MULTILINGUAL
ONTOLOGY DEVELOPMENT**

Ahmed Maher Ahmed Tawfik Moustafa

Advisor:

Professor. Fausto Giunchiglia

Università degli Studi di Trento

Abstract

The world is extremely diverse and its diversity is obvious in the cultural differences and the large number of spoken languages being used all over the world. In this sense, we need to collect and organize a huge amount of knowledge obtained from multiple resources differing from one another in many aspects. A possible approach for doing that is to think of designing effective tools for construction and maintenance of linguistic resources and localized domain ontologies based on well-defined knowledge representation methodologies capable of dealing with diversity and the continuous evolvement of human knowledge. In this thesis, we present a collaborative platform which allows for knowledge organization in a language-independent manner and provides the appropriate mapping from a language independent concept to one specific lexicalization per language. This representation ensures a smooth multilingual enrichment process for linguistic resources and a robust construction of ontologies using language-independent concepts. The collaborative platform is designed following a workflow-based development methodology that models linguistic resources as a set of collaborative objects and assigns a customizable workflow to build and maintain each collaborative object in a community driven manner, with extensive support of modern web 2.0 social and collaborative features.

Keywords

Knowledge Representation, Multilingual Resources, Ontology Development, Computer Supported Collaborative Work

Acknowledgments

I am particularly grateful for my supervisor, Professor. Fausto Giunchiglia, for the guidance and advices he has provided throughout my time as a PhD student. The good times were those when I was directly supervised by him. I would like to thank also all my colleagues of the KnowDive group and in particular those who contributed in this thesis; Vincenzo Maltese for the fruitful discussions and suggestions, Francesco Bux for the strong support in conducting the comparative evaluation. Finally, Amit Kumar, Subhashis Das, and Enrico Nicoletti for participating in the collaborative experiment.

Contents

PART I – INTRODUCTION	1
INTRODUCTION	2
THE CONTEXT	2
THE PROBLEM.....	3
THE SOLUTION	5
INNOVATIVE ASPECTS	6
STRUCTURE OF THE THESIS	7
STATE OF THE ART	9
LEXICAL DATABASES	9
LEXICAL DATABASES ON THE WEB.....	10
ONTOLOGIES.....	12
ONTOLOGY DEVELOPMENT TOOLS ON THE WEB	16
<i>Semantic Wiki based tools</i>	16
<i>Interactive web based tools</i>	18
PART II – METHODOLOGICAL CONTRIBUTIONS	20
UNIVERSAL KNOWLEDGE CORE (UKC)	21
NATURAL LANGUAGE CORE (NLC)	22
CONCEPT CORE (CC).....	25
UKC USAGE SCENARIOS	27
1- <i>Lexical Resource Enrichment</i>	27
2- <i>Lexical Resource Translation</i>	28
3- <i>Ontology Design and Development</i>	28
4- <i>Ontology Localization</i>	29
COLLABORATIVE DEVELOPMENT METHODOLOGY	30
COLLABORATIVE OBJECT	30
COLLABORATIVE WORKFLOW	31
STEPS IN THE METHODOLOGY	35
PART III – TECHNOLOGICAL CONTRIBUTIONS	41
UKC WORDNET APPLICATION	42
USER STUDIES.....	42

REQUIREMENT SPECIFICATIONS	44
USER INTERFACE DESIGN	47
UKC INTERACTIVE PLATFORM	48
USER STUDIES	48
REQUIREMENT SPECIFICATIONS	49
OVERALL ARCHITECTURE	62
USER INTERFACE DESIGN	64
COMPARATIVE EVALUATION (WORDNET VS. UKC).....	69
QUANTITATIVE WORK	69
EVALUATION RESULTS	70
UKC COLLABORATIVE PLATFORM	73
USER STUDIES	73
REQUIREMENT SPECIFICATIONS	75
OVERALL ARCHITECTURE	81
USER INTERFACE DESIGN	83
COLLABORATIVE DEVELOPMENT EXPERIMENT	87
ONTOLOGY DEVELOPMENT METHODOLOGY: DERA FRAMEWORK.....	87
EARLY EXPERIMENT	90
<i>Experimental Setting</i>	91
<i>Results of the experiment</i>	91
FINAL EXPERIMENT	93
<i>Experimental Setting</i>	93
<i>Results of the experiment</i>	95
CONCLUSION AND FUTURE WORK	99
COLLABORATIVE DEVELOPMENT OF ENTITY TYPES.....	99
COLLABORATIVE DEVELOPMENT OF DOMAINS	100
BIBLIOGRAPHY	102
APPENDIX A - INTERVIEWS.....	115

Part I – Introduction

1. Introduction

- **Context**
- **Problem**
- **Solution**

2. State of the Art

- **Lexical Databases**
- **Ontologies and Ontology Development Tools**

Chapter 1

INTRODUCTION

THE CONTEXT

Recently, there have been great advances in semantic-aware and context-aware applications. Semantic-aware applications are mainly intended to assist with information retrieval. They are designed to return more accurate search results by trying to extract the embedded meaning of the search keywords. On the other hand, context-aware applications are smart applications capable of detecting the user's social and physical surroundings (i.e. physical location, weather forecast) and provide in-site recommendations and short answers to user's queries submitted in natural language. Both semantic-aware and context-aware applications rely on knowledge based approaches, i.e. approaches which exploit the semantics of information in order to deliver the necessary and sufficient information to the user and avoid delivering irrelevant and extra information. Examples of knowledge based approaches include: automatic classifications (Giunchiglia, F. et al. 2007 b), abstract reasoning (Giunchiglia, F. et al. 1997), ontology matching (Giunchiglia, F. et al. 2007 a), ontology mapping (Giunchiglia, F. et al. 2010a), common sense reasoning (Bouquet, P. et al. 1995) and natural language data and metadata understanding (Zaihrayeu, I. et al. 2007).

One of the major requirements of knowledge based approaches is to consider the diversity in human knowledge as people in different parts of the world have different ways of living and thinking. Diversity appears in the same natural language terminologies as the same word may refer to more than one object (homonymy) and the same object might be referred to with more than one word (synonymy). A major challenge appearing here is how to deal with diversity in order to increase the accuracy of semantic-aware and context-aware applications. In fact, this requires huge background multilingual linguistic resources which must provide adequate coverage for the diversity of the world and means of transforming this big amount of linguistic data

into useful domain specific knowledge. In addition, it must allow for extensibility as knowledge is continuously evolving.

Therefore, this challenge reflects two main research directions that we need to go through: (1) Defining methodologies for capturing and organizing multilingual linguistic information in a formal way; and (2) Designing and implementing usable tools for gathering diverse linguistic terminologies and cross-culture knowledge.

Capturing and organizing multilingual linguistic information in a formal way can be achieved by working at idiom level, i.e. by defining a piece of knowledge, or a semantic entry and its mapping to a set of lexical definitions in multiple languages forming semantic multilingual linguistic resource. Enriching this linguistic resource requires interactive and user-friendly web application that allows geographically distributed linguistic and domain experts to contribute in a collaborative manner. Their collaboration should take place based on a well-defined collaborative methodology which states clearly the development processes, user roles, and access rights. In addition, the collaborative methodology needs to remove the communication barrier between the participants and allow them to discuss various domain-related topics, share ideas and reach to common agreements.

THE PROBLEM

Knowledge is continuously evolving in a diversified world. Diversity is a world fact that we have to address and deal with; it appears in more than one dimension:

- ***Diversity in natural language terminologies:*** the same object maybe referred to with more than one word. For example, the word fridge is used throughout the northern and western regions of the United States while the word refrigerator is used more throughout the southern and eastern regions. On the other hand, the same word may refer to more than one meaning based on the context of usage. For example, the word kind means sort or type in one sense and means gentle or merciful in another sense.

- ***Diversity in formal language representations:*** any representations may differ according to the scope of the domain, purpose of the representation or the conceptual point of view of the domain expert. For example, one domain expert may classify cars as economic, luxury, and sports cars while another domain expert may classify them based on car models and brands. Another approach is to allow the two experts to collaborate and come up with a classification that capture both point of views.
- ***Diversity in human knowledge:*** knowledge may differ based on cultural differences or personal opinions. For example, Arab countries bring up horses only for riding and racing purposes while other countries may bring them up as a food.

As a result, a complete digital formal linguistic resource which acts as source of knowledge for knowledge seekers and as a background database for semantic applications is very hard to construct and maintain. This is mainly for the following specific reasons: (1) it should be very large and capture most if not all of the real life concepts, (2) it should be accurate which requires huge manual intervention in order to deal with problems like natural languages ambiguity and defining appropriate context for words usages, (3) it should be diversity aware and capable of capturing the regional and cultural differences, (4) it should be well organized and semantic aware in order to be reused in building efficient end-user software applications, (5) it should be extensible since knowledge is continually evolving with time and provide feasible extensible mechanisms without the need to reconstruct any part of the knowledge base, and (6) it should be easily accessible anywhere with friendly user-interface and don't require complicated installation process.

There is a lack of tools that cover these set of features together and provide support for manual collaborative enrichment of linguistic resources and ontology development in a way that promote diversity aware and cross-culture domain knowledge availability.

THE SOLUTION

This thesis aims to provide a methodological and technological approach for designing and implementing a collaborative platform for enriching linguistic resources and developing multilingual ontologies. The platform facilitates the management of diversity across cultures and development of localized domain ontologies.

The platform uses the Universal Knowledge Core as the background methodology for knowledge representation, also referred to as UKC, a framework designed and implemented by knowdive¹ group. The UKC framework defines a comprehensive methodology for organizing knowledge obtained from multiple languages into three main levels: (1) Natural language level, (2) Formal Language Level, and (3) knowledge level (Giunchiglia, F. et al. 2012a and Giunchiglia, F. et al. 2012b). Therefore, the UKC framework provides the necessary background semantic representation we need to rely on.

After deciding on the background semantic representation, we defined a collaborative development methodology based on the notions of collaborative objects and collaborative workflows. The methodology models the project under development into a set of collaborative objects and assigns a customizable collaborative workflow for each object. The collaborative workflows could model the process required for the development of a specific linguistic object and specify the roles of domain experts who will participate in the process, other constraints like time constraints could also be managed by setting timers and due dates to the workflow process. The workflow is customizable since workflows differ from project to project according to the structure and nature of the project. For example, the development of specific domain ontology for banking systems may require development of several classes of objects related to the financial transactions (i.e. withdraw, transfers, and the rest of financial transaction types) and other classes of objects related to banking card types (credit, debit, and the rest of banking card types). The objects related to financial transactions contain a fair amount of sensitive domain specific attributes, therefore the collaborative methodology

¹ <http://disi.unitn.it/~knowdive/>

could be adopted to model these objects as collaborative objects and assign to them a workflow process that requires approval from at least four banking domain experts. On the other hand, Objects related to banking cards types are simpler and well known to domain experts, thus these collaborative objects could be assigned a workflow process that requires approval from only one banking domain expert. This fine grained development methodology may results in more accurate development of domain ontologies and within managed time ranges.

The collaborative methodology has been applied on UKC content and a collaborative interactive web platform has been designed, implemented and evaluated. The web platform was designed and implemented by following an iterative approach started by implementing a multilingual version of WordNet application accessing UKC content as a read-only linguistic resource. In parallel, an initial version of the UKC web platform was designed and implemented to provide an interactive environment for manipulating UKC content as an editable linguistic resource. A comparative evaluation was conducted between the UKC web platform and WordNet application in order to come up with a final refined version. Finally, the UKC web platform was further evolved to support collaborative development of UKC content according to the proposed collaborative methodology. The final collaborative platform allows participants to collaborate together in a social media and build a virtual community by leveraging a group of web 2.0 social features like message boards, online discussions, and interactive polls.

INNOVATIVE ASPECTS

The flexible, yet managed, collaborative approach proposed in this thesis is applicable to any project that requires development in community-based manner. Modelling a project as a set of collaborative objects and assigning a customizable workflow instance to each collaborative object is a fine grained development approach. Most of the commonly existing collaborative development tools offer the “single approval model”, an approach which allows for collaborative development and keeps the changes pending till getting approved by a domain expert.

However, there have been few trials to develop specific workflow-based approaches. (Palma, R. et al. 2008) and (Sebastian, A. et al. 2008) proposed an editorial workflow approach for ontology development using specific workflow ontology for defining workflows. Both approaches define domain specific workflows by making use of predefined workflow ontology. Our approach differs as it defines a set of domain non-specific collaborative workflows and makes use of them on the level of collaborative objects under development instead of one custom workflow for the development of the whole domain.

STRUCTURE OF THE THESIS

The thesis is divided into three parts. Part I introduces the problem statement, the proposed solution and the state of the art. The state of the art explains briefly the notions of lexical databases and ontologies then points out to the common web-based lexical databases and ontology development tools.

Part II introduces the methodological contributions of the thesis. Chapter 3 presents the UKC methodology for knowledge organization and describes how the methodology provides a clear distinction between formal and natural language representations. Chapter 4 presents a novel methodology for collaborative development. The methodology is based on the notions of collaborative objects and collaborative workflows. In this chapter, we explain the methodology and how it could be applied in collaborative development in step-by-step basis.

Part III introduces the technological contributions of the thesis. Chapter 5 presents the UKC WordNet application, a multilingual version of the famous WordNet application that has been designed to access the UKC linguistic resource. Chapter 6 presents the UKC Interactive Platform, a rich client web application for analyzing and enriching the UKC linguistic resource. Chapter 7 presents a comparative evaluation between UKC WordNet application and UKC interactive platform.

Chapter 8 presents the UKC collaborative platform, a web application for UKC linguistic resource enrichment and formal ontology design in a community based manner. Chapter 9 documents a collaborative development experiment conducted in order to

design a multilingual ontology about the “flowing bodies of water”, the experiment took place as part of the evaluation process of the UKC collaborative platform. Chapter 10 concludes the thesis and points out to the future work.

Chapter 2

STATE OF THE ART

In this chapter, we explain briefly the notions of lexical databases and ontologies and point out to the major contributions and related work in these two areas.

We first start by defining the notion of lexical database and explain the requirements of a complete and well defined lexical database. Then we list the available lexical databases currently existing on the web and their main characteristics.

Then we define the notion of ontology and list the available ontology development tools. Our main focus is on ontology development tools that are characterized by being collaborative and web-based.

LEXICAL DATABASES

The word *lexical* is defined in the word reference dictionary² as:

Adjective: “... related to words or the vocabulary of a language as distinguished from its grammar and construction...”

Starting from this definition, we can provide a more formal definition for a lexical database as an organized resource of the vocabulary of a language. It stores information about morphemes, the smallest possible unit of a language such as words, and meanings.

A lexical database could be designed in many different ways but however a complete lexical database should capture the knowledge that a native speaker has about the language. It should store at least the following information:

² <http://www.wordreference.com/>

- Words, word forms, and phrases.
- Meaning of words with usage examples.
- The word part of speech (noun, verb, adjective, or adverb)
- Relations between words and phrases.

Another important distinction should be made when designing a lexical database is regarding its users, *human users* vs. *natural language processing engines*. For human users, it might be sufficient to capture and store the least amount of linguistic details that could be used in designing effective end-user applications. For natural language processing engines, the lexical database must be machine readable which requires defining means of organizing linguistic information in semantic ways and preserve the semantic relations between them.

Lexical databases could be multilingual and store the vocabulary of multiple languages; designing such a database introduces a couple of additional requirements:

- Provide means for matching lexical entries of any two languages. For example, matching synonym lexical entries might be used in building applications like online dictionaries.
- Allow for extensibility and adding more languages in a straightforward manner.

LEXICAL DATABASES ON THE WEB

There are small number of lexical databases and linguistic resources that exist today and have web platforms for linguistic analysis and enrichment. In the following paragraphs we review them briefly:

WordNet (Fellbaum, C. 1998 and Miller, George A. 1995) is a famous and accurate electronic lexical database of English nouns, verbs, adverbs, and adjectives grouped into a set of cognitive synonyms called *synsets*. WordNet lexical database was constructed and organized based on psycholinguistic principles related to theories of human mind lexical organizations. It was developed manually by a group of knowledge experts which

was the main reason for its accuracy. The latest version of WordNet released in November 2012 contains 155,287 words organized in 117,659 synsets.

MultiWordNet³ and **EuroWordNet**⁴ are multilingual lexical databases aligned with WordNet and structured in the same way as WordNet. They are considered as an extension to WordNet which resolve its multilingual drawback. Both of these multilingual resources provide synsets which are strictly aligned with WordNet English synsets and their semantic relations were imported from WordNet and preserved for the translated synsets.

BabelNet (Navigli, R. and Ponzetto, SP. 2010) is a multilingual semantic network constructed automatically following a methodology that integrates lexicographic and encyclopedic knowledge from WordNet and Wikipedia. Its multilingual support was automatically constructed using machine translations in order to enrich their lexical representations. BabelNet wasn't constructed based on a formal representation. BabelNet provides a graphical user interface, known as **BabelNetExplorer** (Navigli, R. and Ponzetto, SP. 2012). BabelNetExplorer allows the users to visually explore the knowledge repository but maintaining and enriching the linguistic resource is not supported.

FrameNet (Baker, Collin F. et el. 1998 and Baker, Collin F. 2003) is lexical database of English has more than 10,000 word senses. It's machine-readable database, based on providing annotated examples of how words are used in actual texts. FrameNet is constructed based on a theory called Frame Semantics which assigns a semantic frame to each concept, i.e. a description explains the usage of the concept and its relations with other concepts. FrameNet has been constructed manually by defining language independent frames and annotated examples. The multilingual lexicalization was done as separate projects.

³ <http://multiwordnet.fbk.eu>

⁴ <http://www.ilc.uva.nl/EuroWordNet>

To the best of our knowledge, none of these tools has been built on a methodology for organizing knowledge obtained from multiple languages into natural language level and formal language level in order to facilitate extensibility and multicultural environment support. We could not also find a tool that provides collaborative extensibility based on a novel collaborative methodology. Table 2.1 provides a comparison between our collaborative platform and the commonly used linguistic tools, our platform supports all the listed distinctive features that none of them supports all of them.

Linguistic Tool	Multilingual Resource	Distinction between Natural and Formal Language	Manual Construction	Content Analysis Using Interactive tool	Collaborative Enrichment
WordNet	No	No	Yes	Yes	No
MultiWordNet	Yes	No	Yes	Yes	No
EuroWordNet	Yes	No	Yes	Yes	No
BabelNet	Yes	Yes	No	Yes	No
FrameNet	Yes	Yes	Yes	Yes	No
Our Tool	Yes	Yes	Yes	Yes	Yes

Table-2.1: Comparison between our tool and the commonly used linguistic tools.

ONTOLOGIES

Ontology is a hybrid term used in philosophy and information sciences. The origin of the word comes from the Latin word *Ontologia* (ont- + -logia), which means the *science of existence*.

The word *ontology* is defined in the word reference dictionary as:

(1) **Noun:** “... the branch of metaphysics that studies the question of what it means to exist...”

(2) **Noun:** “...the set of entities presupposed by a theory...”

Starting from this definition, we can simply state that an ontology aims to define theories and models about reality and relations between existing objects. There have been several definitions for the notion of ontology in philosophy and information sciences. In order to stay in line within the context of this thesis, we focus only on the definitions proposed by information scientists.

In 1993, Gruber defined the notion of ontology as an “explicit specification of a conceptualization” (Gruber, T. 1993). In other words it means a clear well-defined representation of an abstract view of the world. A conceptualization is merely an abstraction of the world, or part of the world, in the form objects and relationships between objects.

In 1997, Borst proposed another definition and defined ontology as a “formal specification of a shared conceptualization” (Borst, W. N. 1997). Borst has added the word “shared” to describe the conceptualization which means that the abstract representation should be based on a common agreement instead of a personal point of view. Borst also replaced the word “explicit” with another word “formal” which means that the abstract representation should be free from ambiguity.

In 1998, Studer merged these two definitions in one definition: “ontology is a formal, explicit specification of a shared conceptualization.” (Studer et al 1998).

The three definitions agreed on defining an ontology as a well-defined conceptualization. For instance, let’s consider the field of computer science as an example of an ontology (Figure 2.1), an ontology engineer may organize it into set of concepts and relations. The concepts include the topics such as Artificial Intelligence, Natural Language Processing, and Machine Learning. The relations between them include subtopic-of, prerequisite-to and complementary-to.

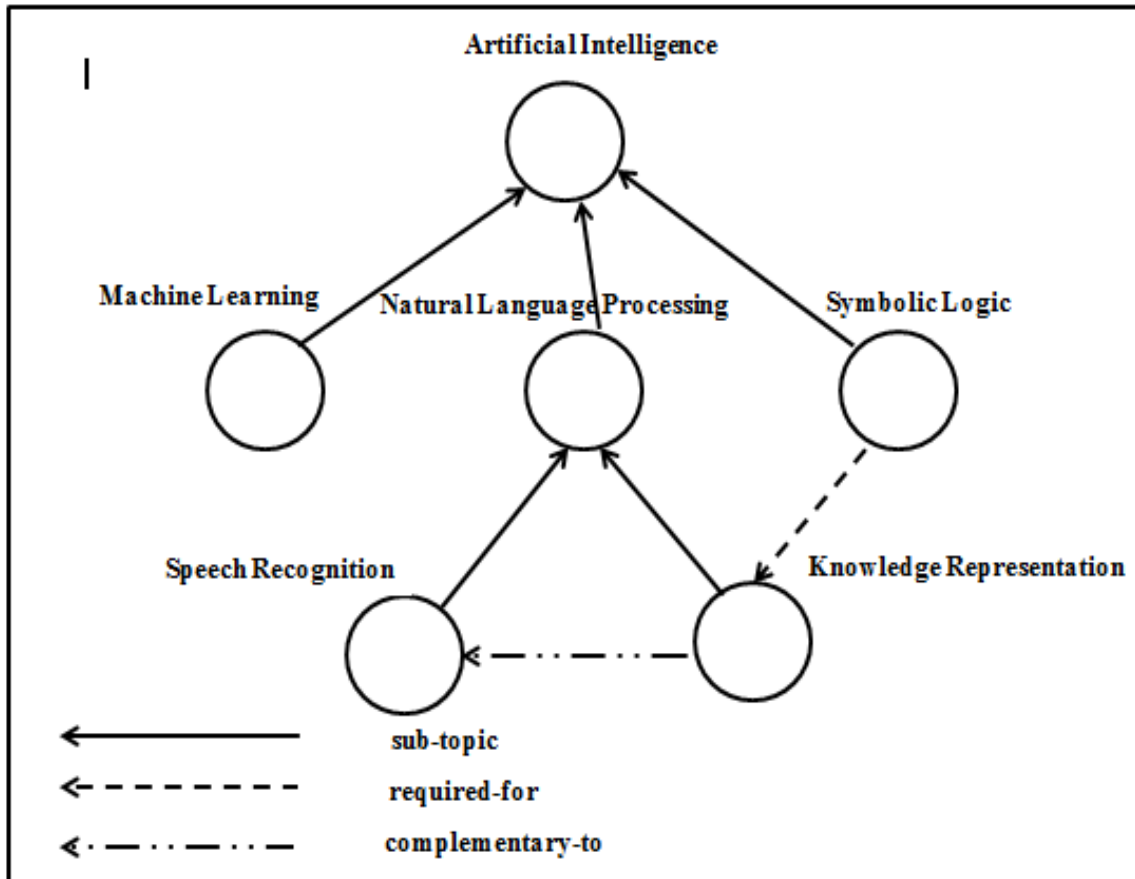


Figure-2.1: Simplified Computer Science Ontology

There have been also several trials to classify ontologies into different types. (Meersman et el 1999) classified ontologies into two main categories:

- Thesaurus: domain specific ontology that models object, facts, and relations of a certain domain such as computer science, economics, or law.
- Lexicon: a language specific ontology that models object vocabularies of specific language. Lexical databases defined earlier are examples of lexicon ontologies.

Another classification proposed by Giunchiglia (Giunchiglia et el. 2009b). This paper introduced the notion of lightweight ontologies as the backbone structure of the ontology visualization graph which is composed of concepts labelled with natural language names (nodes) and semantic relations (edges).

Lightweight ontologies are classified into two categories:

- ***Descriptive lightweight ontologies:*** define the meaning of terms as well as the nature and structure of a domain.
- ***Classification lightweight ontologies:*** describe and classify large collection of data items.

After defining and classifying ontologies, it's essential to understand the benefits of developing new ontologies. Since ontologies provide a shared conceptualization and common understanding of a specific domain of interest, they may play a major role in supporting information exchange in various application areas.

(Fensel, D. 2001) identified three main application areas where ontologies may be used as the background schema for information exchange:

1. ***Information Search and knowledge management:*** these application areas include end-user applications and organization-wide applications.
2. ***Web Commerce:*** these application areas include online shopping applications.
3. ***Electronic business:*** these application areas include enterprise and e-government applications.

Since ontologies are used nowadays in several application areas, one common pitfall is to think of an ontology as being equivalent to a database schema or to consider them analogous. There are two main differences between an ontology and a database schema: (1) an ontology is designed based on a common agreement for the purpose of knowledge representation and sharing while a database schema could be modelled on individual basis for the purpose of data storage, and (2) an ontology is represented in a formal language supporting real-life reasoning while a database schema is modelled in a simple tabular form.

ONTOLOGY DEVELOPMENT TOOLS ON THE WEB

Ontology development tools that are web based and allow for collaborative development are few. Our main focus is on web based tools that support collaborative ontology editing through the web. Tools like *OntoStudio* (Weiten, M. 2009) and *TopBraid Composer* (COMPOSER, T. 2007) are mature tools for ontology development and offer wide range of functionalities such as ontology modeling, importing ontologies in standard formats, and ontology matching. Both of them offer collaborative development extension to the main tool which doesn't go beyond internal shared access of a locally controlled ontology. Therefore, it would be hard to adopt them as collaborative tools for building diversity aware ontologies and require participation from a geographically distributed domain experts while web based tools could be easily adopted for this purpose. Web based ontology development tool used these days could be classified into two main categories: (1) Semantic Wiki based tools; and (2) Interactive web based tools.

SEMANTIC WIKI BASED TOOLS

Semantic Wiki aims to combine traditional wiki systems with Semantic Web by introducing semantic web technologies like RDF and OWL to the traditional Wiki. Ontology development tools that are based on Semantic Wiki based systems have gained popularity during the past few years with the increase of active contributors to the traditional encyclopedia⁵ as they could be easily extended and become familiar to many domain experts. Semantic Wiki based tools are capable of providing a development approach similar to the one provided by the traditional wiki. Although the development process is straight forward but it is not customizable based on the contents and the nature of domain under development. On the following paragraphs we go through the main and the most promising Semantic Wiki based ontology development tools.

⁵ <http://www.wikipedia.org/>

OntoWiki (Auer, S. et al. 2006) is a knowledge base which provides visual representation of domain ontologies as information maps. Information map entries are represented as web accessible pages and interlinked to related digital resources. Ontowiki also provides contextual views for entities, i.e. map views for locations and calendar views for instance data. The tool supports collaborative content enrichment by enabling users to rapidly editing or adding contents through an inline editing mode analogous to the WYSIWYG (What You See Is What You Get) editing strategy for text editing, since information can be edited in the same environment as it is presented to users. Social collaboration features are supported by OntoWiki: (1) commenting on contents, (2) tracking all changes performed by contributors such as: contributions to the ontology schema, additions of entities or comments, and information about the contributor; and (3) entities rating

CofficientMakna (Tempich, C. et al. 2007) is a Semantic Wiki tool for collaborative ontology engineering that allows participants to create ontologies from scratch or import existing ontologies to the wiki, imported ontologies are mapped to the wiki hypertext model according to a predefined schema. The collaborative development is augmented with the use of an argumentation ontology that formalizes the arguments exchanged between participants (issues, ideas and discussions) and provides a reasoning mechanism can alert users if they agree and disagree on the introduction of the same ontology entity.

MoKi (Ghidini, C. et al. 2010) is a tool for modeling ontologies and enterprise process models in a collaborative MediaWiki⁶ based approach. The tool associates a wiki page containing both unstructured and structured information to each entity of the ontology. The unstructured information contains the MediaWiki markup format (text, images, drawings, or any markup format) while the structured information contains description knowledge stored according to the modeling language adopted (RDF or XML) where each entity is described by means of triple having the form (subject, relation, object). Moki supports multi-mode access to the page contents and exchange of comments to provide ease of use for different categories of users such as domain experts and knowledge engineers.

⁶ <http://www.mediawiki.org/>

INTERACTIVE WEB BASED TOOLS

Ontology development tools that are designed and implemented as interactive web based tools are sort of challenging. They require more facilities in addition to exposing the ontology development functionalities. They should provide means for ontology visualization and manipulation in interactive way, management of concurrency control, mechanisms for data storage and alignment. On the following paragraphs we go through the main and the most promising interactive web based development tools.

OntoLingua (Farquhar, A. et al. 1997) was one of the earliest trials to provide collaborative ontology development on the web. It supports collaborative ontology construction by providing simultaneous work tasks through group sessions, i.e. a user opens a session and then may assign another group of people ownership to it. This enables any other member of that group to join the session and work simultaneously on the same set of ontologies. One of the main drawbacks stands in the fact that it was built using outdated web standards, for instance the server cannot notify users that a change has occurred until they revisit the page again. This tool also has no social collaboration features.

Protégé (Tudorache, T. et al 2007 and Tudorache, T. et al 2008a) is an open-source community with a suit of plug-ins that allow domain experts to construct domain models and knowledge-based applications using ontologies. Protégé supports the creation, manipulation and visualization of ontologies in various formats (RDF, OWL, and XML). The Protégé platform supports two main ways of modelling ontologies; Protégé frames editor models ontologies as a set of classes organized in a subsumption hierarchy to represent a domain's fundamental concepts and a set of slots associated to classes to describe their properties and relationships, Protégé-OWL editor models ontologies for the semantic web using the Web Ontology Language (OWL). ***WebProtégé*** (Tudorache, T. et al 2008b and Tudorache, T. et al 2012) is an extension project that supports collaborative ontology editing through the web. It allows multiple users to edit the same ontology at the same time and all changes made by one user are seen immediately by other users with possibility of adding comments and annotations. In addition, it has support for web 2.0 features such as discussions, online chatting, and

voting on content. Collaborative Protégé has an extension for supporting project specific workflows that could be defined using a generic ontology for modelling workflows (Sebastian, A. et al 2008a and Sebastian, A. 2008b). A workflow execution engine is required to interact with Protégé to run the modelled workflow for a specific project. This approach differs from our approach, as the workflow is modelled for a specific project and our approach offers a customized workflow for each collaborative object in the project.

Table 2.2 provides a comparison between our collaborative platform and the commonly used ontology development tools, our platform supports all the listed distinctive features that none of them supports all of them.

Linguistic Tool	Content Analysis Using Interactive tool	Collaborative Enrichment	Ontology Localization	Web 2.0 Features	Workflow Support	Workflow Per Collaborative Object
MoKi	No	Yes	No	Yes	No	No
OntoWiki	No	Yes	No	Yes	No	No
CofficientMakna	No	Yes	No	Yes	No	No
OntoLingua	No	Yes	No	No	No	No
Protégé	Yes	Yes	No	Yes	Yes	No
Our Tool	Yes	Yes	Yes	Yes	Yes	Yes

Table-2.2: Comparison between our tool and the commonly used ontology development tools.

Part II – Methodological Contributions

1. Universal Knowledge Core

- **Natural Language Core (NLC)**
- **Concept Core (CC)**

2. Collaborative Development Methodology

- **Collaborative Object**
- **Collaborative Workflow**
- **Steps in the Methodology**

Chapter 3

UNIVERSAL KNOWLEDGE CORE (UKC)

The Universal Knowledge Core (UKC) is a comprehensive and extensible multilingual knowledge base. One of our main goals is to build the UKC as a high quality cross-culture and diversity aware knowledge base. In order to achieve this goal, UKC has been designed based on a knowledge organization methodology which keeps it unique from existing linguistic resources. For instance, WordNet is among the most widespread and commonly used lexical resources. However, it has several limitations. In particular, it's only in British English, and the glosses given for the terms reflect the British society and culture. For example, the term *"primary school"* is defined as *"a school for young children; usually the first 6 or 8 grades"* which is clearly biased towards the British educational system. Thus, as it is, WordNet in a multilingual and multicultural environment, it does not bring a real competitive advantage. On the other hand, UKC provides a mapping from word forms (coming from different languages) to language independent concepts in order to represent and analyze how a language independent concept is expressed in different languages, or which synonymous sets are used to express that meaning in each language, or a lexical gap in case a lexicalization for the concept doesn't exist in the language. Apart from this lexical mapping, the UKC manages also semantic relations between concepts which allows for developing domain specific ontologies in a language independent manner.

UKC is made of different cores, but in this thesis we are focusing only on two of them, Natural Language Core (NLC) and Concept Core (CC). In this chapter, we first start by explaining the Natural Language Core then the Concept Core and finally we provide four possible usage scenarios and techniques for enriching the UKC framework and ontology development.

NATURAL LANGUAGE CORE (NLC)

Words are the basic linguistic units of languages. Each word in a natural language may have one or more meanings, known as word senses. One word may have over than 20 meanings based on the context of the word usage in the sentence. For instance, the word *play* has a different meaning in each of the following sentences:

- “We went to see the *play* Romeo and Juliet at the theatre”
- “The coach devised a great *play* that put the visiting team on the defensive”
- “The children went out to *play* in the park”
- “Can you play on this old recorder?”

The Natural Language Core (NLC) models a language as a huge container of *synsets* and *lexical gaps*. A synset is a set of words having the same word sense. A synset, in addition to being a set of synonym words, is also characterized by having a natural language gloss and a part of speech (POS). The gloss is a brief description of the meaning of the words, it provides a brief explanation of the words with one or more usage examples. The part of speech (POS) is the category to which a word belongs in accordance to its syntactic functions, it indicates whether a word is either noun, adjective, verb, or adverb.

Every synset is associated with one language-independent concept. It may happen that a language-independent concept has a corresponding synset in one language and a gap in another language, formally known as a *lexical gap*. This model is similar to WordNet as it groups words with same meaning into synsets but it extends WordNet as it deals with multiple languages and accounts for lexical gaps.

Figure-3.1 gives an example of the English word “kind” which has two different synsets. The first synset is associated with three senses (form, sort, and kind) which correspond to the meaning of “Category of things”. The second synset is associated with one sense (kind) which corresponds to the meaning of “Someone having or showing tender or helpful nature”. In the same figure, the Italian word “Gentile”, the translation of word kind, has one word sense and one corresponding synset.

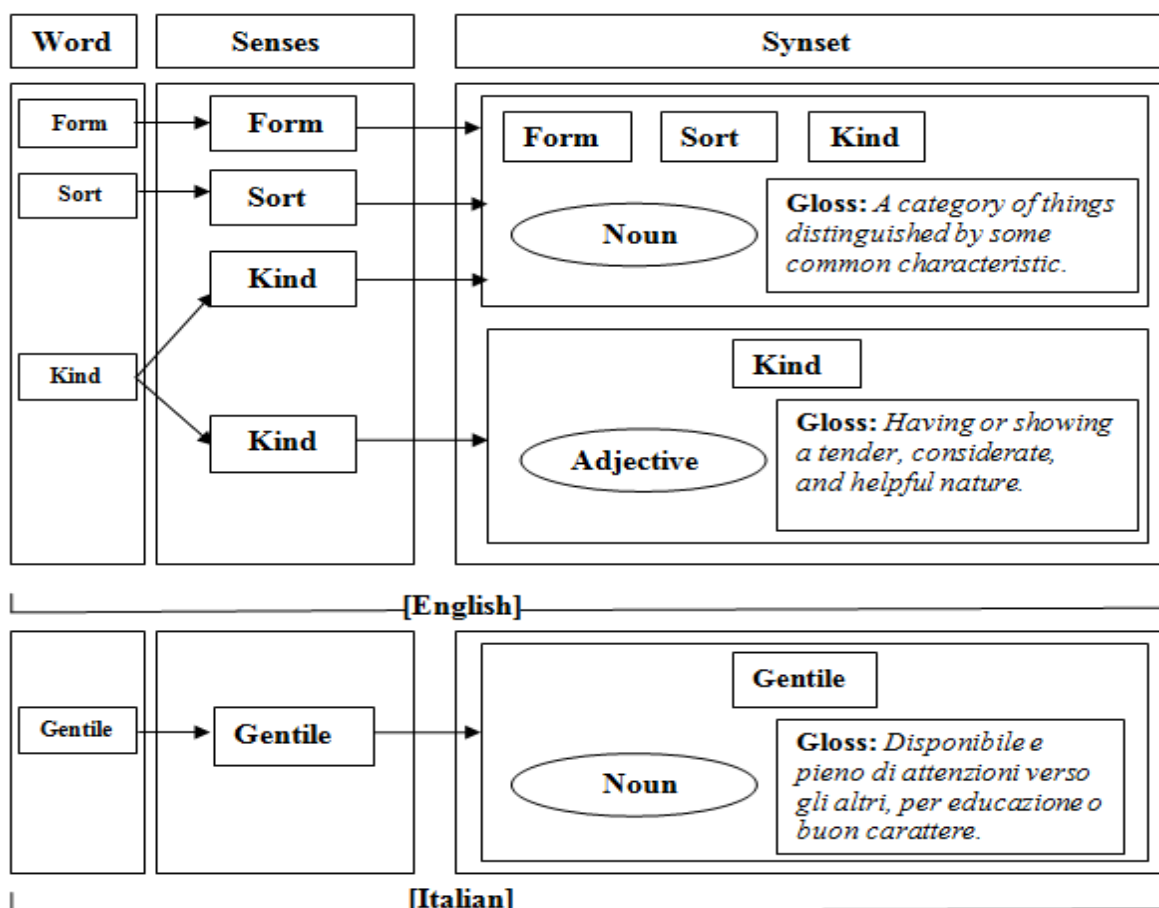


Figure-3.1: The relation between the English word “kind”, its senses, and synsets and the Italian word “Gentile”, its sense and synset.

Synsets and their word senses relate to other Synsets through lexical relations. Natural language core defined 17 different types of lexical relations, here we explain them briefly:

1. **Synonymy:** a symmetric relation connecting two senses having the same Part of Speech and sharing the same meaning, i.e. early synonym soon.
2. **Antonym:** a symmetric relation connecting two senses having the same Part of Speech but having an opposite meaning, i.e. early antonym late.
3. **Homograph-of or homonymy:** a relation between two words having the same Part of Speech and they are spelled or pronounced exactly the same way but they differ in meaning; i.e. bank as financial institution is homograph of bank as sloping land.

4. ***Pertains-to:*** a relation between an adjective and an adjective or a noun, i.e. fraternal pertains to brother.
5. ***Pertainym-of:*** a relation between an adverb and the corresponding relational adjective; chemically pertains to chemical.
6. ***Participle-of-verb:*** relation between an adjective - the participle - and a verb; i.e. breaking is the participle of the verb break.
7. ***Also-see:*** a relation between two adjectives or between two verbs similar in meanings; i.e. unaware and unconscious is similar in meanings.
8. ***Similar-to:*** a relation between two similar adjectives but not equivalent; i.e. ponderous similar to heavy.
9. ***Related-form or derivationally related form:*** a relation between two words where one of them is a lexical variation of the other; i.e. personhood is related to person.
10. ***Hyponymy or subordination:*** a relation between two words indicating subordination or belonging to a lower ranks or classes; i.e. chicken is hyponymy of bird.
11. ***Hypernym or insubordination:*** a relation between two words indicating insubordination or belonging to a higher rank or class; i.e. bird is a superordinate class of chicken.
12. ***Meronymy:*** a relation between a part and the whole; i.e. finger is part of hand.
13. ***Holonymy:*** a relation between a whole and its parts; i.e. hand has finger.
14. ***Substance meronymy:*** a relation between a constituent and a compound; i.e. hydrogen is a substance of water.
15. ***Member meronymy:*** a relation between a member and the belonging group or set; i.e. a football player is a member of a football team.
16. ***Lexical entailment:*** a relation between two verbs denoting activities which are in temporal inclusion; i.e. snoring entails sleeping.
17. ***Verb-group:*** a relation between two verbs representing actions or states which are similar in meaning; i.e. survive is a verb group with exist.

CONCEPT CORE (CC)

The Concept Core (CC) is the codifying information about *language-independent concepts* and relations between them. Every synset in a natural language is associated with exactly one language-independent concept. Each concept is having a concept Id as a unique identifier and a concept label as a descriptive word obtained from the first language-dependent synset associated with the concept. Figure-3.2 associates two language-independent concepts to the language-dependent synsets of (fail and go wrong) and (fail and breakdown) in English and Italian, together with the corresponding semantic relations between those concepts. Given that, the English synset for (fail and breakdown) corresponds to a lexical gap in Italian.

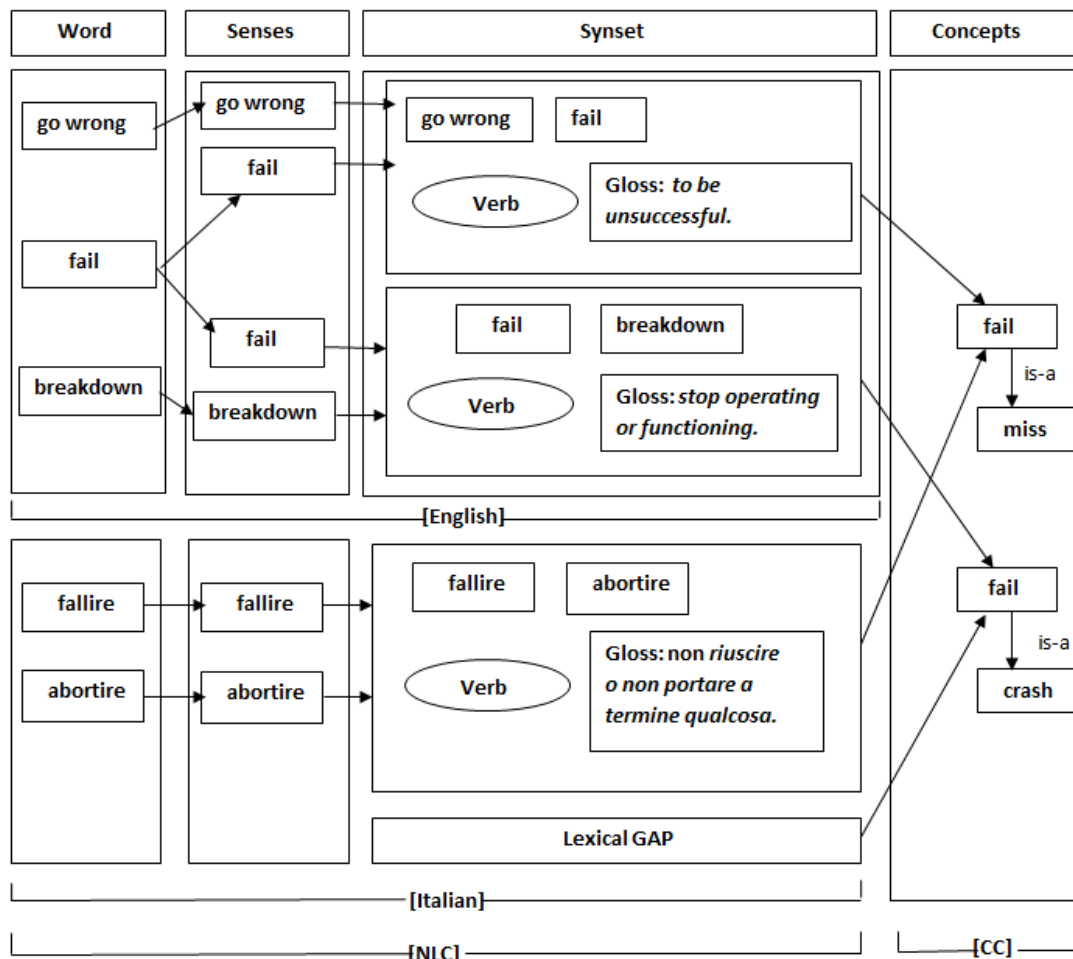


Figure-3.2: The relation between the English word “fail”, its senses, and concepts and the Italian word “fallire”, its sense and concepts.

Concepts are related to other concepts through *semantic relations*. Concept Core has divided semantic relations into two different types of relations: *hierarchical relations* and *associative relations*. Hierarchical relations are transitive, asymmetric, and form coherent hierarchies in the form of parent-child relations, while associative relations connect concepts in different hierarchies in the Concept Core

The Concept Core defined four different types of hierarchical semantic relations. Here we explain them briefly:

1. ***Is-a***: a specialization relation between two concepts that indicates the necessity of specialization, i.e. minivan is-a car.
2. ***Has-aspect***: a relation indicating the possibility of specialization into a subordinate class in a given context; i.e. chicken is food when cooked.
3. ***Part-of***: a relation that holds between a part and the whole; i.e. finger is part of hand.
4. ***Value-of***: a relation that holds between a value and its attribute name; i.e. red is a value of color.

The Concept Core defined three different types of associative semantic relations. Here we explain them briefly:

1. ***Substance-of***: a relation that holds between a constituent and a compound; i.e. hydrogen is a substance of water.
2. ***Member-of***: a relation that holds between a member and the belonging group or set; i.e. football player is a member of a football team.
3. ***Metaphor-of***: a relation that holds between the metaphoric meaning and its literal meaning; leg of a table is a metaphor of leg of a human.

UKC USAGE SCENARIOS

In this section, we define the possible scenarios that could be accomplished using UKC. There are mainly four possible scenarios for enriching UKC:

1. Enriching UKC as a multilingual linguistic resource.
2. Translating a linguistic item exists in UKC from one language to a new language.
3. Designing language-independent formal ontologies using language-independent concepts.
4. Localizing existing formal ontologies to a new language.

1- LEXICAL RESOURCE ENRICHMENT

Preparation :

- (1) Choose a target language to enrich.
- (2) Define the new synset (words, gloss, and POS)
- (3) Check if the defined synset has a language-independent concept.



Enrichment:

- (1) If the language-independent concept has been found, the new synset is mapped to it. Otherwise, a new language-independent concept and a new synset are added to the lexical resource.

2- LEXICAL RESOURCE TRANSLATION

Preparation :

- (1) Choose a source language (language to translate from).
- (2) Choose a target language (language to translate to)
- (3) Obtain the synset which require translation from the source language and it's language-independent concept.



Translation:

- (1) Define the translated synset in the target language (words, gloss, and POS) or a lexical gap if the target language has no corresponding synset.
- (2) Map the translated synset to the obtained language-independent concept.

3- ONTOLOGY DESIGN AND DEVELOPMENT

Preparation:

- (1) Choose a target language
- (2) Analyze the domain that should be captured by the ontology.
- (3) Define the ontology conceptual model.



Development:

- (1) Retrieve the ontology language-independent concepts.
- (2) For missing concepts, define new concepts and their corresponding synsets in the target language.
- (3) Define semantic relations between language-independent concepts.

4- ONTOLOGY LOCALIZATION

Preparation:

- (1) Choose a source language (language to translate from).
- (2) Choose a target language (language to translate to).
- (3) Obtain the ontology language-independent concepts.
- (4) Obtain the corresponding ontology in the source language.



Localization:

- (1) Define the translated synsets in the target language (words, gloss and POS) or a lexical gap if the target language has no corresponding synset.
- (2) Map translated synsets to the ontology language-independent concepts.

Chapter 4

COLLABORATIVE DEVELOPMENT METHODOLOGY

Recently with technology advances such as web 2.0 and the widespread of social networks and online communities, Ontology development has been transformed from a process performed by a single domain expert in isolation into a collaborative process performed by a group of domain experts distributed all over the world.

In this chapter, we explain our methodological approach for collaborative enrichment of multilingual linguistic resources and development of ontologies. The methodology incorporates web 2.0 features to the development process through the use of collaborative objects and collaborative workflows.

The chapter starts by defining the notions of collaborative objects and collaborative workflows, highlighting on the differences between collaborative and non-collaborative objects and between collaborative workflows and standard process management workflows. The chapter then proceeds by explaining the proposed collaborative methodology in step by step basis and applying the methodology for UKC development as a practical case study.

COLLABORATIVE OBJECT

A *collaborative object* is a web-based item that could be instantiated in a collaborative manner. Examples of well-known collaborative objects include web based online meetings and social events organized using shared online calendar. The main difference between a collaborative and non-collaborative object stands in the fact that the former needs to be defined based on common agreement. Table 4.1 provides a brief comparison between the characteristics of collaborative objects and non-collaborative objects.

	Collaborative Object	Non-collaborative Object
Value is assigned as a result of collaborative effort?	Yes	No
Value could be known in advance?	No	Yes
Value could be calculated based on specific rule or mathematical formula?	No	Yes

Table 4.1- Collaborative Object vs. Non-collaborative Object

COLLABORATIVE WORKFLOW

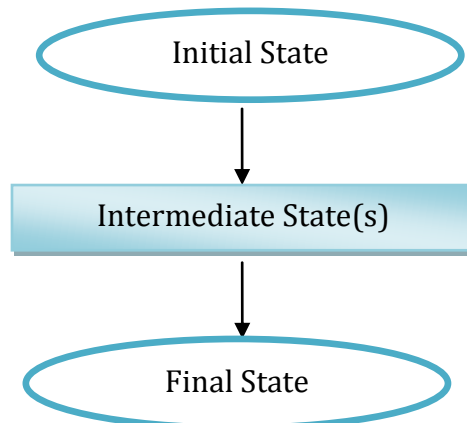
In order to explain clearly the concept of a collaborative workflow, we initially start by defining the *standard process management workflow* as an automation of a work process during which the work tasks are passed from one participant to another according to predefined set of rules and each participant is assigned a specific user role. An efficient process design and implementation should result in an improved work process and elimination of any unnecessary steps. The standard workflow process is designed and implemented using workflow management software. On the other hand, we define the *collaborative workflow* as an automated process implemented using workflow management software augmented with social collaboration software (online discussions, interactive polls, or any other collaborative software tool). The collaboration software is introduced in order to facilitate communication via facilities supporting discussions and exchanging of ideas among the participants. The collaborative workflow is expected to provide significant efficiency gains to the process after eliminating the communication barriers between the participants and transforming the single-user decision making steps into common decision agreement steps. Table 4.2 provides a brief comparison between the two types of workflows.

	Collaborative Workflow	Process Management Workflow
Instance could be fully automated without human intervention ?	No	Yes
Instance may have a predefined input, if fulfilled the instance completes successfully ?	No	Yes

Table 4.2- Collaborative Workflow vs Process Management Workflow

The main requirement for supporting a collaborative workflow is to provide social collaboration facilities and a work breakdown structure of an automated process. The work breakdown structure is provided in the form of different types of process nodes and user roles that are meant to constitute the main structure and sequence of workflow process steps. In our collaborative methodology definition, we are going to define and use six different types of nodes. A node can be *a state, a human task, a condition, a fork, a join, a timer, and a notification*. Each node has a unique set of properties; we explain them briefly as follow:

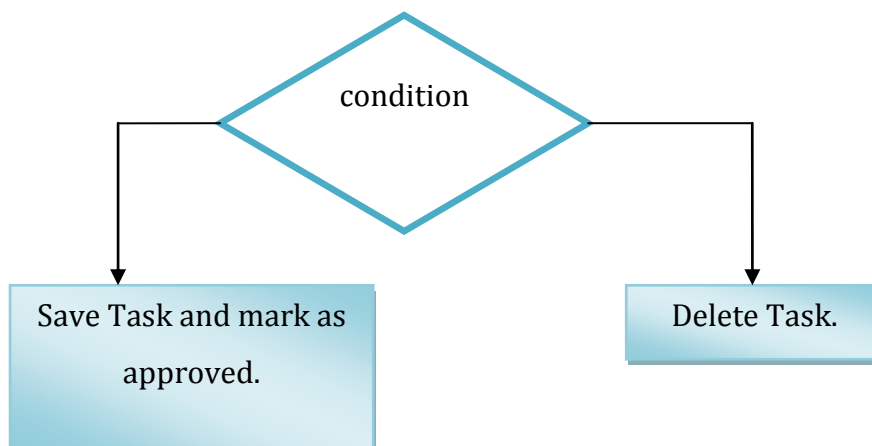
1. **A state:** represents a step in the workflow process that executes immediately and requires no user intervention. Any workflow process starts with an initial state and terminates with a final state.



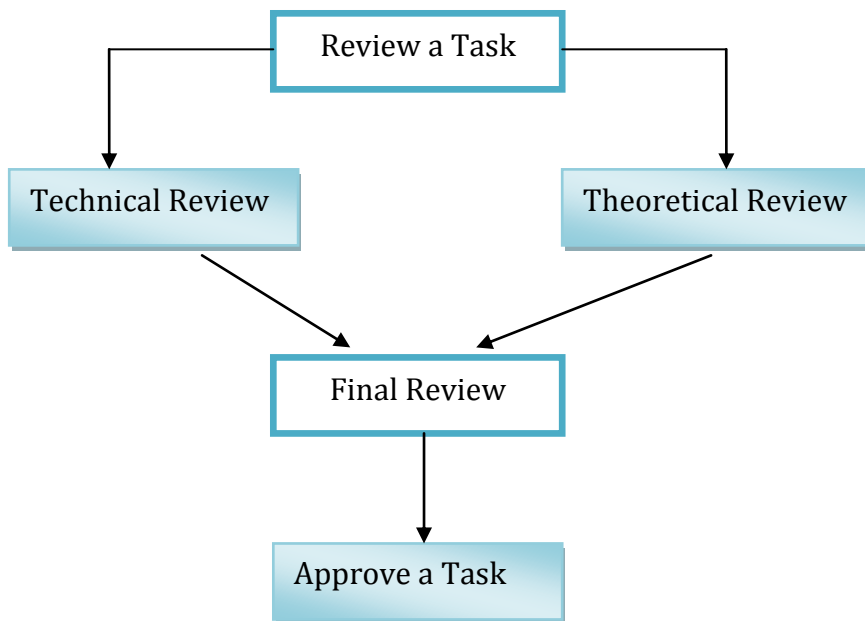
2. **A task:** represents a step in the workflow process that requires human intervention or input from a user. The task is blocked until user input completes. Tasks should be linked with defined user roles or user groups sharing a common role, i.e. a user who can complete a task review must be holding the reviewer role.



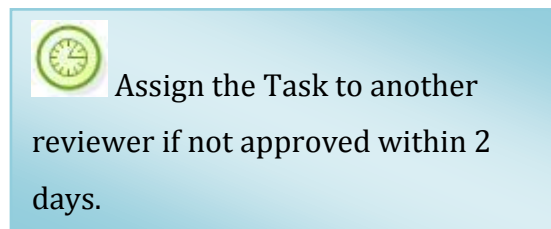
3. **A condition:** represents a decision making step and based on the condition the workflow takes a specific route.



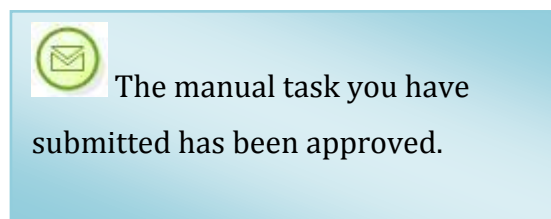
4. **Fork and Join:** are used together to model parallel processing in a workflow process. The fork node splits the flow into two parallel sub-flows in order to perform parallel processing tasks. The join node merges them back and retains the original workflow only in case of successful completion of the two parallel sub-flows.



5. **A timer:** assigns a specific duration for tasks in order ensure that important tasks in a workflow aren't forgotten or left undone for a long period of time due to absence of users.



6. **A notification:** is a message sent to one of the participants to communicate a piece of information.



STEPS IN THE METHODOLOGY

In this section, we propose the collaborative methodology in step-by-step basis. The methodology is not limited to UKC ontology development and localization, it is generic enough to be applied to any collaborative development project. In fact, we are considering UKC as a practical case study in order to explain and verify the methodology. The methodology is composed of three steps, here we explain them:

1- Definition of the collaborative objects: this step requires modeling the project as collection of collaborative and non-collaborative objects.

In UKC (Figure 4.1), four collaborative objects were defined: *Synset*, *Lexical Gap*, *Semantic Relation*, and *Lexical Relation*. The *Concept* object is the only non-collaborative object as it is auto-generated with the creation of a new synset which has not yet associated with already defined concept. In other words, the first language defining a new notion triggers the generation of the language-independent concept.

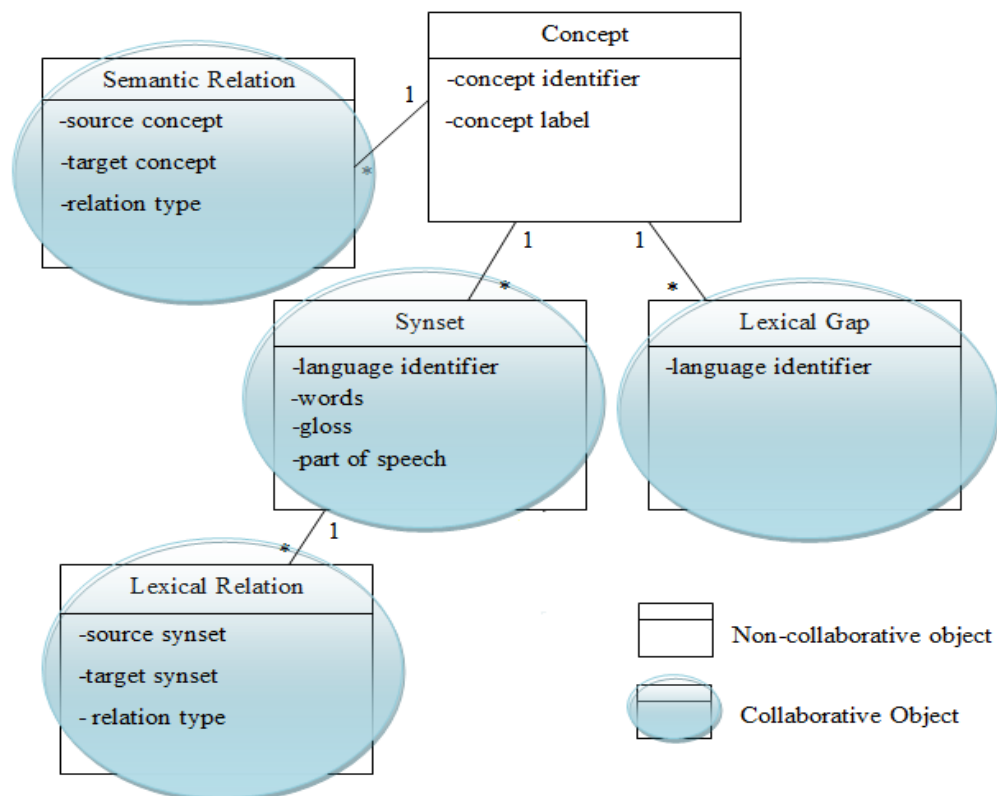


Figure 4.1- Collaborative and non-collaborative objects in UKC

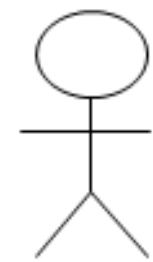
2- Definition of the collaborative workflows: this step requires defining the user roles and the collaborative processes needed to manipulate the collaborative objects. We may define one or more collaborative workflows based on the structure and nature of the project under development. The complexity of the defined workflows may vary based on the nature of domain under development and the number of participants required.

In UKC, the collaborative process is a two-phase process (development phase and validation phase) which required two main user roles: *UKC Developer* and *UKC Validator*. We then define and explain briefly two types of workflows containing two human tasks involving UKC Developers and Validators: (1) *The single approval workflow*, and (2) *The group approval workflow*.


The single approval workflow process (Figure 4.2) is instantiated when a participant holding UKC Developer role manipulates a collaborative object via CUD operation (Create, Update, or Delete). The workflow process then assigns a validation task to a participant holding UKC validator role. The validator decides whether to accept or reject the developer's recent manipulation. In case of rejection, the task is sent back to the same developer for reviewing and resubmission. While taking into consideration that the whole process augmented with social collaboration features (comments, discussions boards, and other features) and the final decision is taken as a result of common agreement between the participants.

The group approval workflow (Figure 4.3) is similar to the single approval workflow except that the approval decision is taken by a group of validators based on a specific condition, i.e. at least three out of four validators (majority vote) should accept the manipulation.

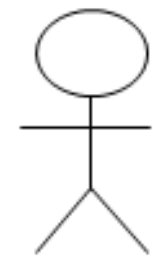
UKC Developer



Manipulate Collaborative Object

 System assigns a new validation task to the corresponding UKC Validator and sends a notification to the validator


UKC Validator



Validator views and validates that task.



No

 Add a rejection reason and notify the developer to resubmit the task.

Yes

The task is marked as approved and the updated entries become available to End Users

Single Approval Workflow

Figure 4.2: Single Approval Workflow

UKC Developer

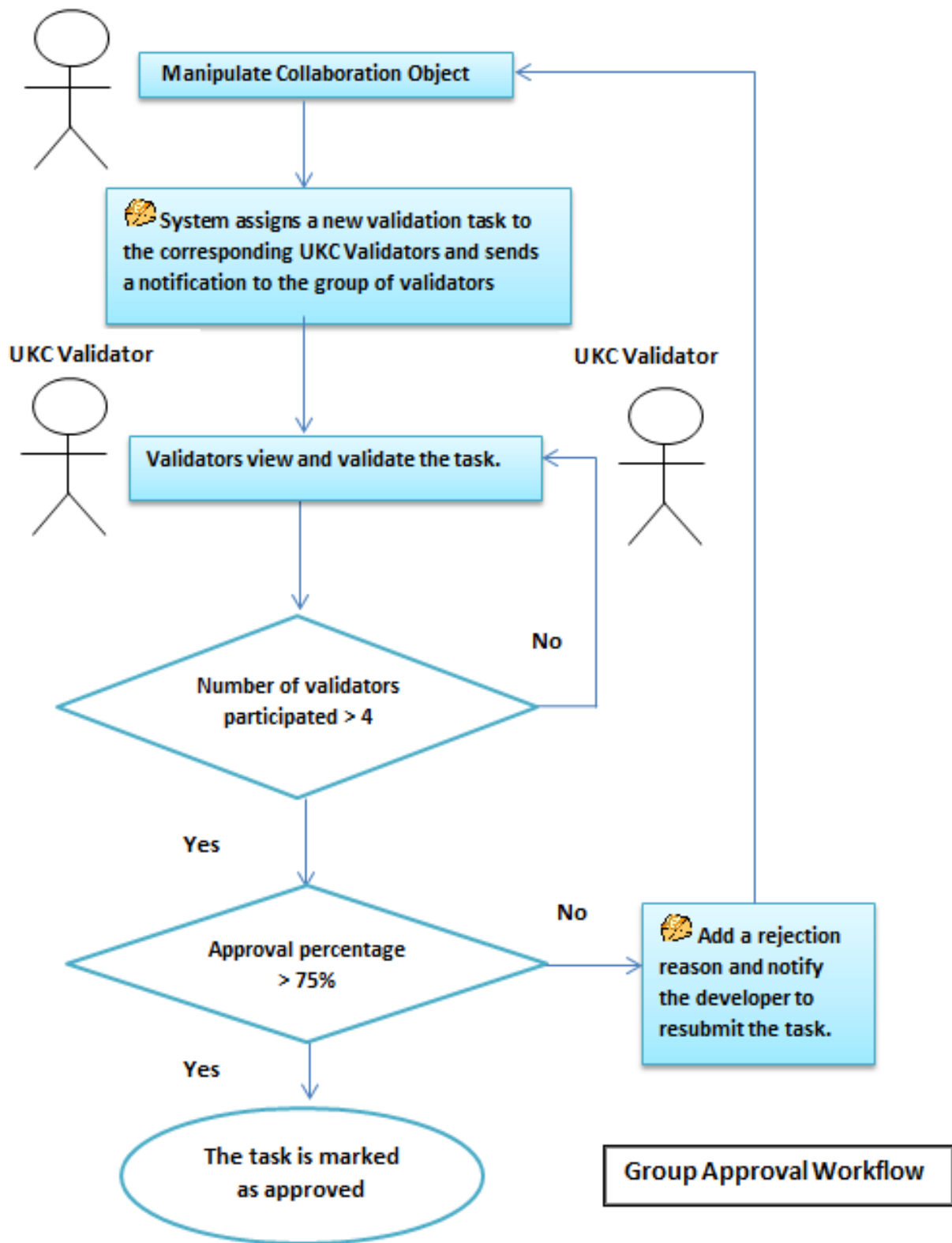


Figure 4.3: Group Approval Workflow

3- Mapping objects to collaborative workflows: this step requires specifying which collaborative workflow needs to be employed for each collaborative object. However, such mapping can be partial and it is not mandatory to map all the collaborative objects to collaborative workflows. In fact, unmapped collaborative objects are directly manipulated without going through any process management procedure.

In UKC (Figure 4.4), we mapped the development of synsets to the group approval workflow while the development of lexical gaps is mapped to the single approval workflow and the rest of collaborative objects are not mapped to any collaborative workflows

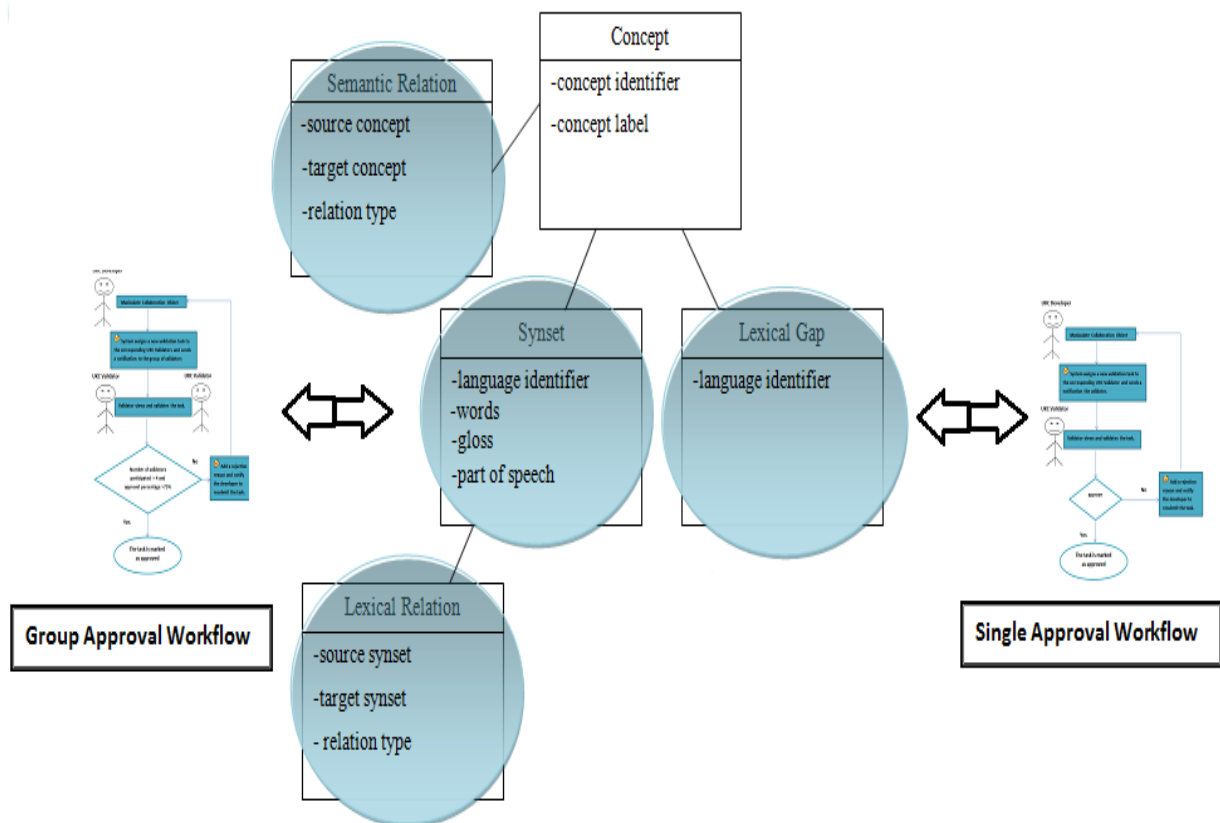


Figure 4.4: Mapping UCK Collaborative Object to Collaborative Workflows.

Figure 4.5 summarizes the main phases of the collaborative methodology. We argue that the proposed methodology is fine grained and highly customizable which is expected to provide a high level of accuracy and time saving as well. A practical example for applying this methodology in translating synsets from English to French may reflect the expected advantages. The French language is characterized by being content rich and its synsets may involve a high degree of ambiguity. Therefore, it would be desirable when translating synsets from English to French to confirm the translated synsets using the group approval workflow and after getting at least three approvals from French linguistic experts. On the other hand, a lexical gap in French could be simpler and easy to spot by a French native speaker. Therefore, using the single approval workflow which requires participation of one French linguistic expert could be sufficient to confirm the existence of the lexical gap.

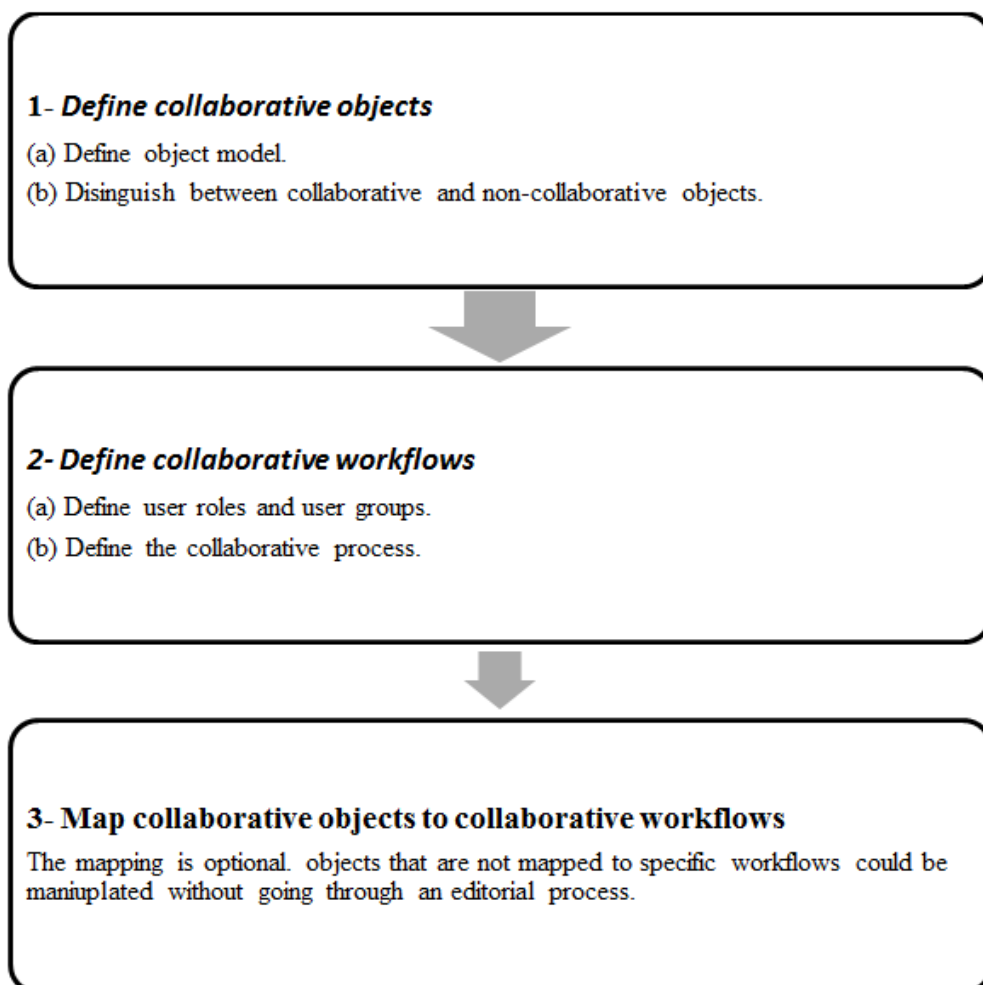


Figure 4.5: The three main phases of the collaborative methodology

Part III – Technological Contributions

1. UKC WordNet Application

- **User Studies**
- **Requirement Specifications**
- **User Interface Design**

2. UKC Interactive Platform

- **User Studies**
- **Requirement Specifications**
- **Overall Architecture**
- **User Interface Design**

3. UKC Collaborative Platform

- **User Studies**
- **Requirement Specifications**
- **Overall Architecture**
- **User Interface Design**

Chapter 5

UKC WORDNET APPLICATION

UKC WordNet is a web application providing access to the linguistic data stored in the UKC framework.

An online demo is available at: <http://uk.disi.unitn.it:8089/ukcui/wordnetview.htm>

The application has been designed to resemble the original user interface of WordNet (<http://wordnetweb.princeton.edu/perl/webwn>) after taking permission from the WordNet team; we sincerely thank them for that. However, the UKC WordNet View differs from the original WordNet in being a multilingual linguistic resource.

The main idea behind designing an application that resembles the original WordNet is to fulfil the following goals: (1) Allow users who are familiar with WordNet to start accessing the UKC and rapidly get to know about its contents and main features, and (2) Conduct a comparative evaluation between our UKC interactive platform and WordNet application while both accessing the same linguistic data source, the comparative evaluation is documented in the next chapter.

USER STUDIES

At the start of any project, it's highly desirable to start by doing: (1) user studies, and (2) requirement specifications. Those are essential steps needed to guide during the design and implementation process.

User studies define the categories of users who are going to access and use the application. Also define the user roles as well as the application views that each user is allowed to access. There are several research methods used to conduct user studies⁷. Here, we explain the most common and relevant ones:

⁷ <http://www.usability.gov/how-to-and-tools/methods/user-research/index.html>

1- Personas: define reliable and realistic representations of the application main users and audiences. Personas should describe users briefly together with their backgrounds, goals and values. They also should try to express and focus on the major needs and expectations of the users.

2- Scenarios: describe the context behind why a specific user or a user group comes to use the application. They explain briefly the user goals and define the possibilities of how the user(s) can achieve them. Scenarios can be represented in several ways; elaborated story-telling scenarios and use case scenarios are the most common ways.

3- Task Analysis: is a process of learning about ordinary users by observing them in action to understand in detail how they perform their tasks and achieve their intended goals. Task analysis helps identify the main tasks that your application must support and also helps application designers refine or re-define specific application features such as navigation or search types by determining the appropriate content needed by actual users.

4- Interviews: are conversations with a purpose between an interviewer and an interviewee. The main purpose of this type of interviews is to gain deeper understanding of the users or user groups who come to use the application. Interviews might take place in individual bases where the interviewer asks the interviewee a predefined set of questions or in groups, known as focus groups, similar to a moderated discussion between at least five or more participants.

5- Online Surveys: are a form of structured questionnaire that targets the application users over the internet generally through filling out a web form. Online surveys can vary in length and format. The collected data can be analyzed to gain insights about users, their goals, and what sort of information users are looking for from using the application.

UKC WordNet view application is a read-only application that was designed for navigating the UKC linguistic resource. We performed the user studies by conducting interviews with knowledge experts and reading the online documentation published by the WordNet team. (Sample of the interviews are listed in Appendix A). The user studies revealed that WordNet has only one user, we called the user: *UKC User*, who is

able to navigate the application, read UKC synsets in a specific language and perform linguistic analysis. In the next section, we present the gathered requirement specifications for UKC WordNet view Application.

REQUIREMENT SPECIFICATIONS

Requirement specifications capture the main system requirements and describe the different forms of interaction modalities between the system and the users.

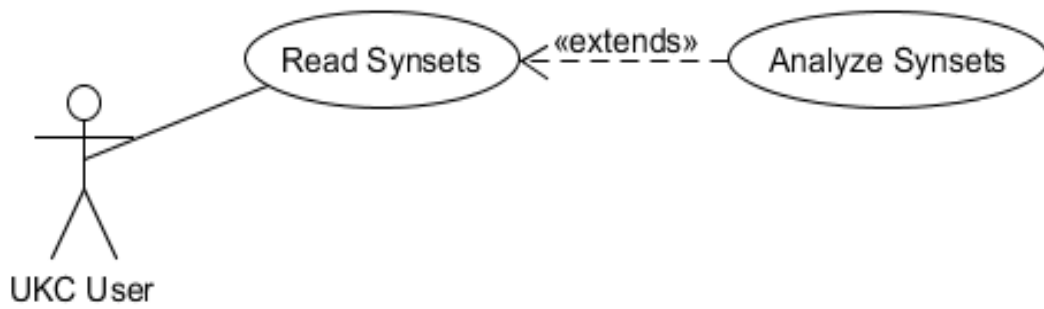
We define the system requirements for the UKC WordNet application in the form of use case scenarios following the standard specification of the unified modeling language (UML)⁸. Use case scenarios are considered as one of the standard techniques for capturing the system functional requirements in the form of a narrative interactive scenario between the user and the system. The scenarios explain the sequence of interactions, user roles, and the system responsibilities. There are mainly three different types of scenario:

- **Main Scenario:** a basic scenario represents a single unit of meaningful work that terminates by achieving a user goal, such as read synset, create synset, or any other possible scenario.

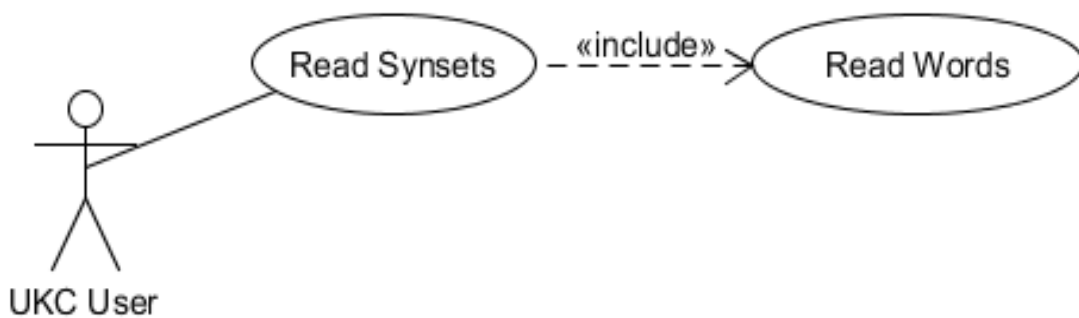


- **Extension Scenario:** an optional scenario that may happen as an extension to the main scenario. For example, analyzing synsets after retrieving them from the system database is an extension scenario.

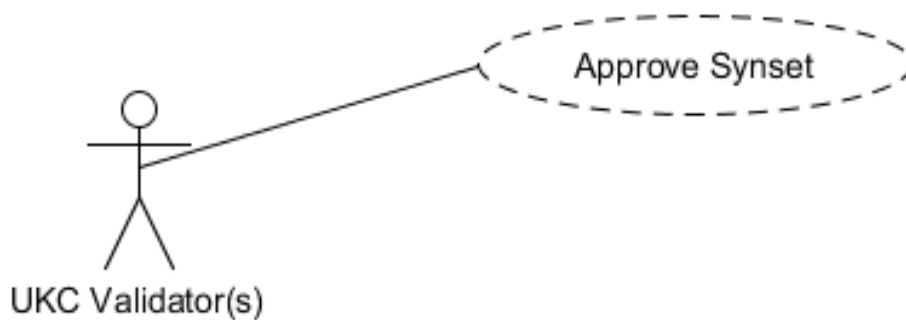
⁸ <http://www.omg.org/spec/UML/>



- **Sub-Scenario:** an internal or included scenario that takes place as a part of the main scenario. For example, reading words is a sub-scenario of reading synsets.



- **Collaborative-Scenario:** a shared scenario that takes place in a collaborative manner between a group of participants. For example, the approval of a newly created synset by a group of knowledge experts.



UKC WordNet application has only one scenario, reading synsets using textual search by a word lemma.

1- Read Synsets

Main Scenario: Read Synsets

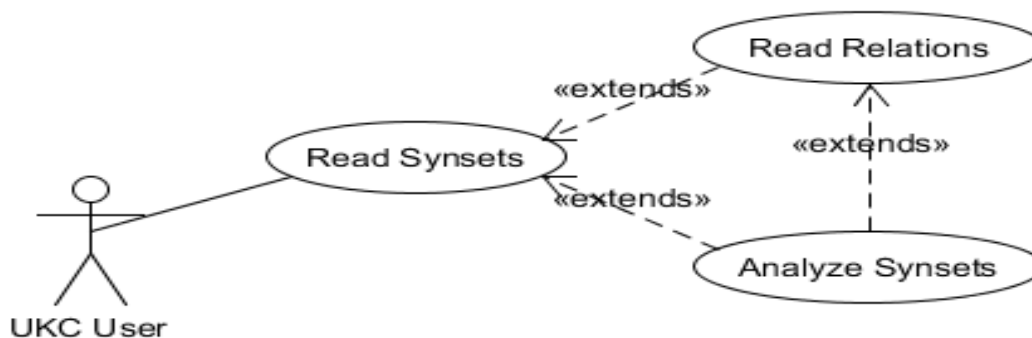
- 1- User chooses the source language and types any word aiming to retrieve the word synsets.
- 2- System auto-suggests words based on what the user has typed.
- 3- User selects one of the auto-suggested words or presses the Search button.
- 4- System retrieves and displays the synsets having a sense for the entered word.

Extension Scenario-1: Read Relations

- 5- User chooses to read the synset relations (semantic, lexical, or semantic-lexical relation)
- 6- System retrieves the synset relations and visualizes them in the vicinity of the original synset.

Extension Scenario-2: Analyze Synsets

- 5- User may decide to analyze the displayed synsets by updating how synsets are displayed (show/hide the gloss or the example sentences).
- 6- System applies the performed visual changes on the displayed synsets
- 7- User can retrieve more synset relations and continue performing the analysis on the new results.



USER INTERFACE DESIGN

Figure 5.1 shows the application user interface. The user can start a new search by typing a word and choosing the desired language. The application then retrieves the set of synsets organized by their part of speech (Noun, Adjective, Verb, and Adverb) and sorted by their synset rank.

The user may interact with the retrieved synsets by clicking on the letter “S” next to the synset to show either semantic relations with the language-independent concept or the lexical relations with language dependent synset. On the other hand, clicking on the letter “W” shows lexical relations with words.

The user can modify the visual display for the displayed synsets by updating the display options. For each synset, it’s possible to show or hide the: gloss, example sentences, synset provenance details, or the language-independent concept identifier for the concept associated with the synset.

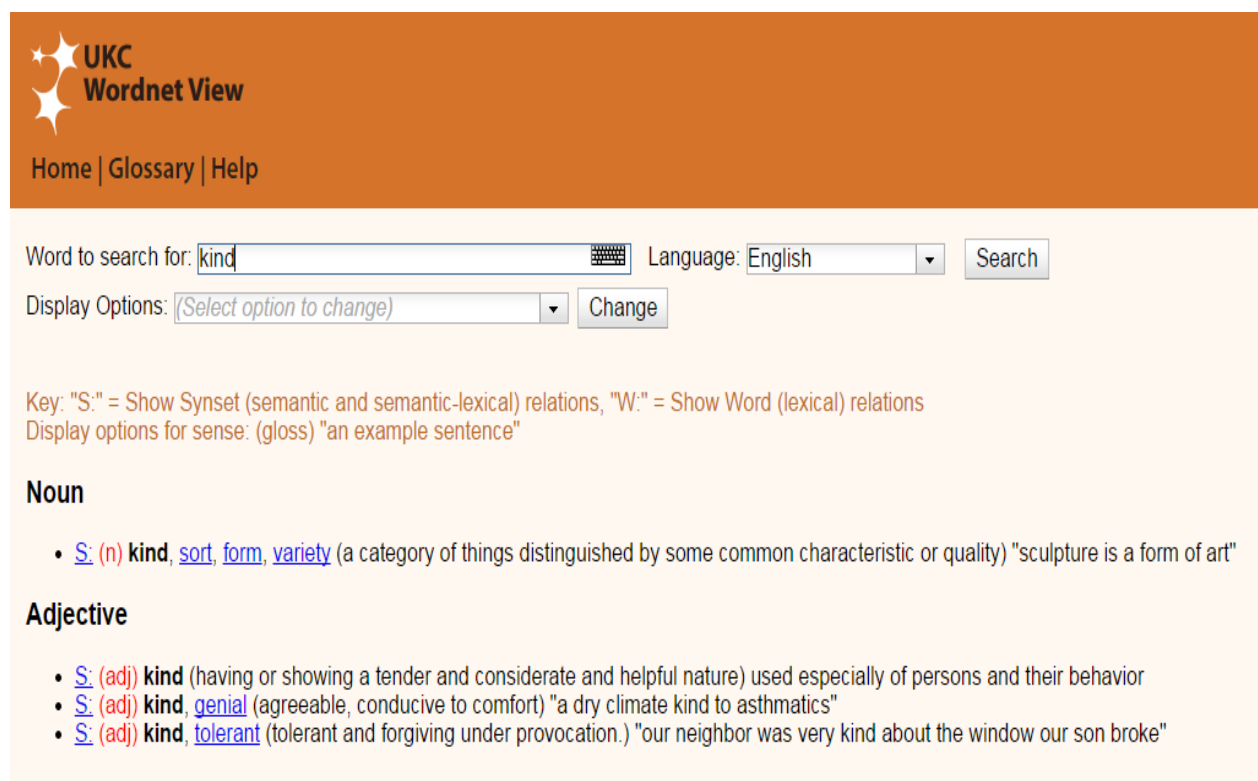


Figure 5.1 UKC WordNet View Application User Interface

Chapter 6

UKC INTERACTIVE PLATFORM

UKC interactive platform is a web application for accessing and manipulating the linguistic data stored in the UKC.

An online demo is available at: <http://uk.disi.unitn.it:8089/ukcui/ukc.htm>

The application was designed and implemented as a rich client internet application having the same instant responsive behavior, look and feel of desktop applications.

In this chapter, we explain the user studies, requirement specifications together with the application overall architecture and user interface design of the UKC interactive platform.

USER STUDIES

User Studies was the first step in designing UKC application as an interactive platform for navigating and enriching UKC linguistic resource. we performed the user studies by conducting interviews with knowledge experts who are fully aware and familiar with linguistic resources. The user studies revealed that that UKC interactive platform has three main categories of users (Figure 6.1):

- **UKC Users:** are those who are allowed to view the UKC knowledge base for the navigation and linguistic analysis without performing any updates to the linguistic resource.
- **UKC developers:** are knowledge experts and domain developers who are able to perform development and maintenance operation on the UKC knowledge base. The UKC developers are allowed to access the application Read/Write interactive features through the application user interface.

- **UKC Administrators:** have full access to system features and they can perform administrative tasks such as creating users, creating groups and assigning user roles to users.

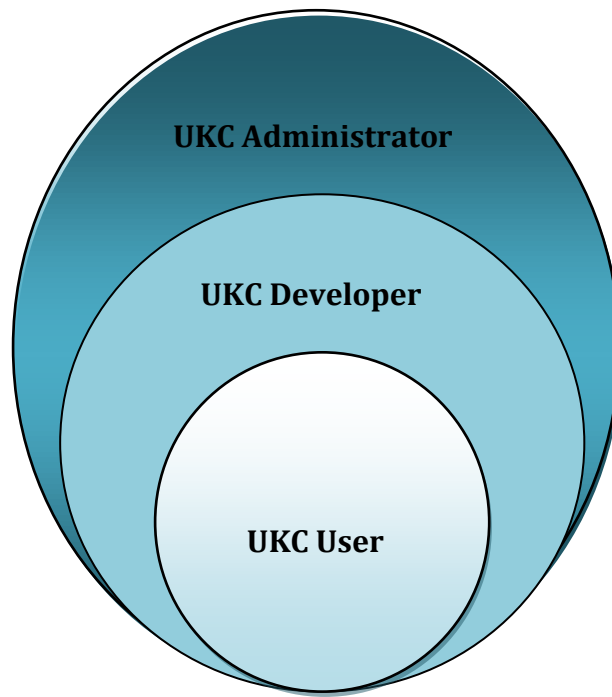


Figure 6.1- Access rights for different categories of users; End Users have read only privileges, UKC Developers have read/write privileges; and UKC Administrators have full access.

REQUIREMENT SPECIFICATIONS

The Natural Language Core (NLC) and Concept Core (CC) scenarios presented in this section explain the core functional scenarios for the management of UKC linguistic resource items: *synsets*, *lexical gaps* and *concepts* as well as *lexical relations* between senses and *semantic relations* between concepts. Knowing that, *words* and their exceptional *word forms* are not managed explicitly through their corresponding synset. For each synset, it is allowed to define any number of synonym words and manage the order of these words.

The UKC linguistic resource is a multilingual one, therefore the user is dealing with two languages:

- ***Working language***: the default language, when the user performs any linguistic read/write operation, system applies the changes on the selected working language.
- ***Reference language***: the alternative language, it's mainly used for multilingual support. For example, it is useful in understanding the meaning of a concept when no lexicalization is available in the working language yet by displaying its corresponding synset in the reference language.

The interactive system should allow the user to switch between the two languages (or change them to other languages) at any moment, or swap the roles of the two language. For examples, a user can start by selecting English as a *working language* and Italian as a *reference language*. In this case, the user will navigate the system in English and perform linguistic operations such as creating new synset for a concept in English. The user may view the corresponding synset associated with the same concept in Italian by selecting to view the lexicalization in the *reference language*. In case the user would like to create a new synset for the same concept in Italian, the user has to update the *working language* to be Italian and selects another language as a *reference language*.

In the following subsections, we explain the development scenarios for UKC linguistic items; synstes, lexical gaps, concepts, lexical and semantic relations. The UML use case scenarios present the requirements and the constraints arise while performing Create-Read-Update-Delete (CRUD) operations and how the system provides interactive features to facilitate these operations and also how the system should prevent the user from performing any pitfall.

UKC interactive platform has 13 basic use case scenarios, here we explain them briefly accompanied with use case diagrams.

1- Read Synsets in the Working Language

Main Scenario: Read synsets in the Working Language

- 1- User types any word in the working language aiming to retrieve the word synsets.
- 2- System auto-suggests words based on what the user has typed.
- 3- User selects one of the auto-suggested words or presses Enter as soon as he/she finishes typing.
- 4- System retrieves and displays the synsets having a word sense for the entered word. For every synset, the system displays the words having the same sense sorted by sense rank, Part of Speech (POS), Gloss, and Examples.

Sub Scenario-1: Read synsets in the Reference Language

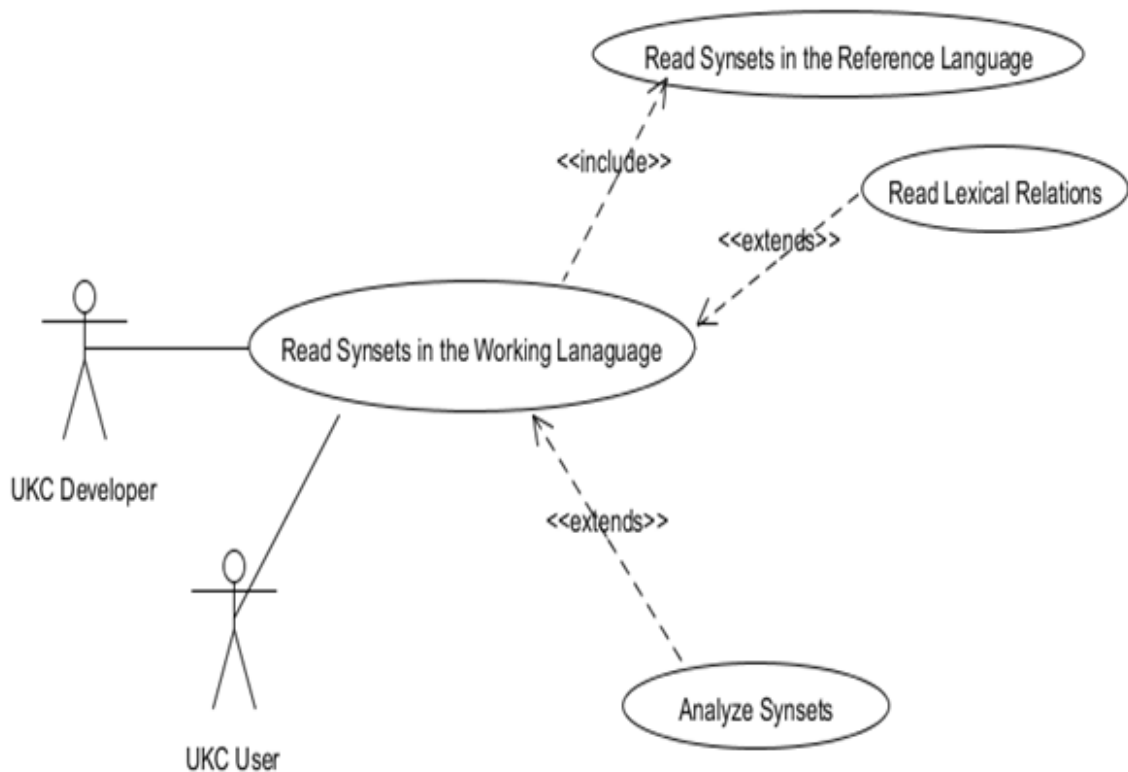
- 5- For each synset retrieved in the main scenario, system retrieves the corresponding lexicalization in the reference language, System retrieves one of the following items depending on the content of the linguistic resource:
 - a- The corresponding synset in the reference language.
 - b- Lexical gap.
 - c- No information, in-case the lexicalization is not yet defined.

Extension Scenario-1: Read lexical relations

- 6- User selects to retrieve the related synsets for any of the retrieved synsets in the main scenario.
- 7- System retrieves the lexically related synsets sorted by relation type.

Extension Scenario-2: Analyze Synsets

- 8- User may decide to analyze the displayed synsets by updating how synsets are displayed (show/hide the POS, gloss or the example sentences) or filtering displayed synsets by POS or lexical relation type.



2- Read the Concept of a Synset

Main Scenario: Read the Concept of a Synset

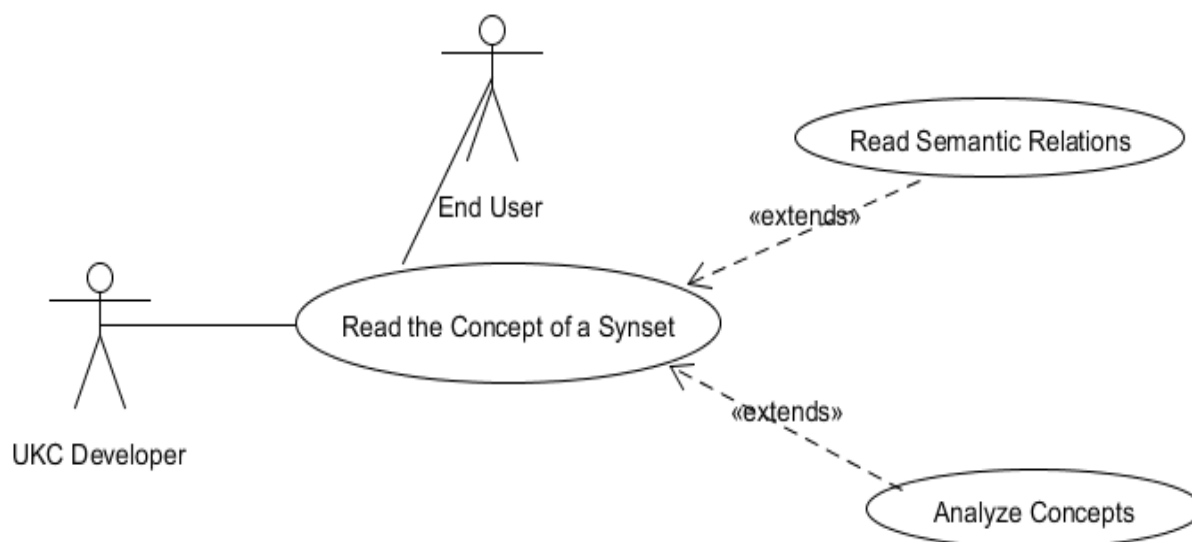
- 1- User retrieves the synsets of any natural language word by following the scenario of use case: [Read Synsets in the Working Language](#).
- 2- User selects to view the concept associated with one of its synsets.
- 3- System retrieves and visualizes the concept. For each concept, the system retrieves its concept identifier and concept label.

Extension Scenario-1: Read Semantic Relations

- 4- User selects to retrieve the related concepts for the retrieved concept or one of its child concepts.
- 5- System retrieves the semantically related concepts sorted by relation type.

Extension Scenario-2: Analyze Concepts

6- User may decide to analyze the displayed concepts by filtering displayed concepts by semantic relation type.



3- Read the Lexicalization of a Concept

Main Scenario: Read the Lexicalization of a Concept

1- User retrieves any language-independent concept by following the scenario of use case: [Read the Concept of a Synset](#).

2- User selects to view the lexicalization of the retrieved concept in the *working language*.

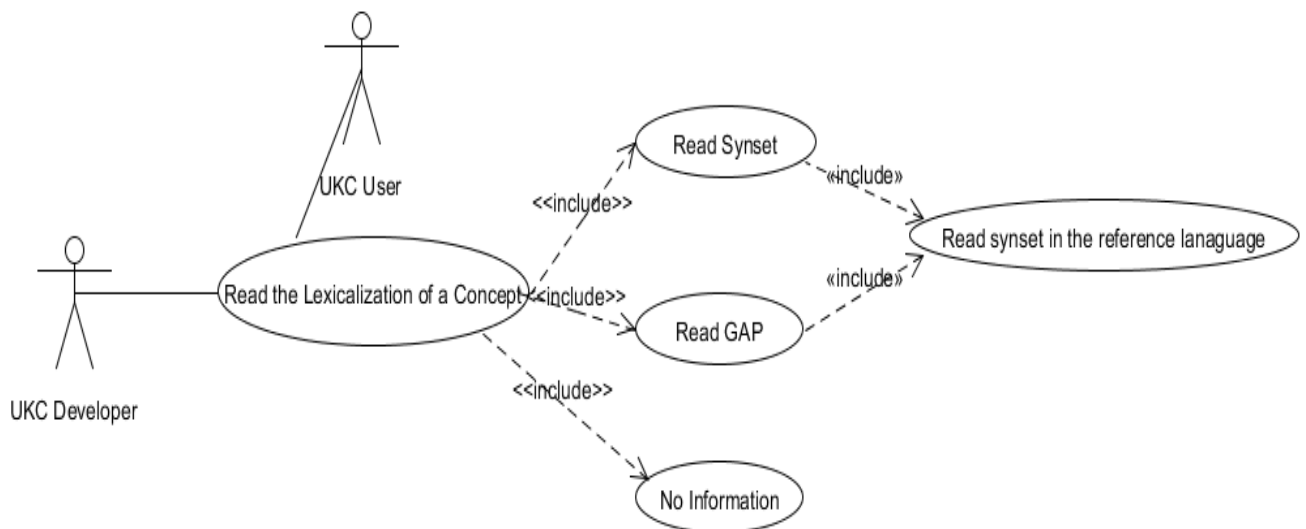
3- System retrieves **one** of the following items depending on the content of the linguistic resource:

- a- The corresponding synset in the working language.
- b- Lexical gap.
- c- No information, in-case the lexicalization is not yet defined.

Sub Scenario-1: Read synsets in the Reference Language

4- For the retrieved synset in the main scenario, System retrieves the corresponding lexicalization in the reference language, System retrieves one of the following items depending on the content of the linguistic resource:

- a- The corresponding synset in the reference language.
- b- Lexical gap.
- c- No information, in-case the lexicalization is not yet defined.



4- Create New Synset with a new Concept

Main Scenario: Create New Synset together with new a Concept

1- User selects to create a notion as a new synset in the working language together with its new language-independent concept.

2- System displays an interactive editable form for defining the new synset and highlights the mandatory fields.

3- User fills the interactive form with the new synset details. The form requires the following mandatory fields:

- Set of word senses.
- Part of Speech (POS).

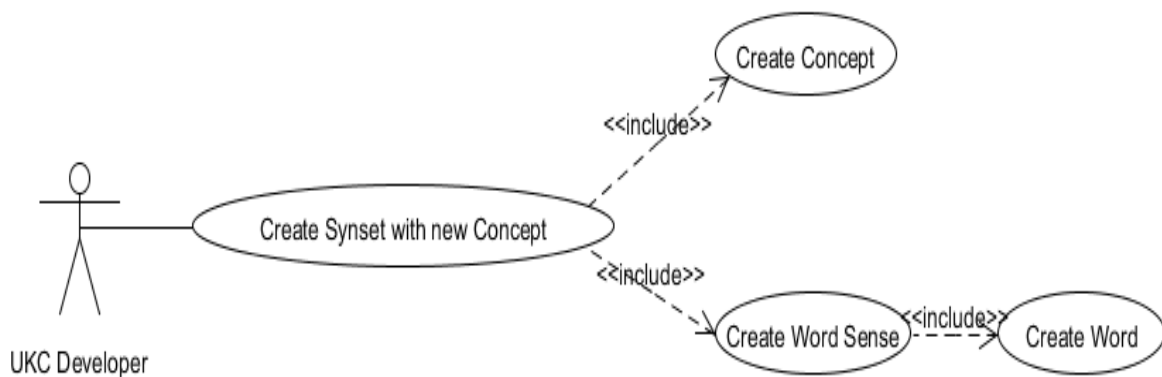
- Gloss
- Example(s)

4- User saves the changes.

5- System checks if the new synset includes new word senses. If it contains any new word sense, system creates the new word sense and stores the word in the linguistic resource.

6- System creates a new synset and automatically creates a new concept and generates the corresponding concept identifier. The concept label is the first word sense defined for the newly created synset.

7- The system associates the newly created concept with the newly created synset.



5- Create New Synset for an existing Concept

Main Scenario: Create New Synset for an existing Concept

1- User selects to create a new synset in the working language together with a new Concept.

2- System displays an interactive editable form for defining the new synset and highlights the mandatory fields.

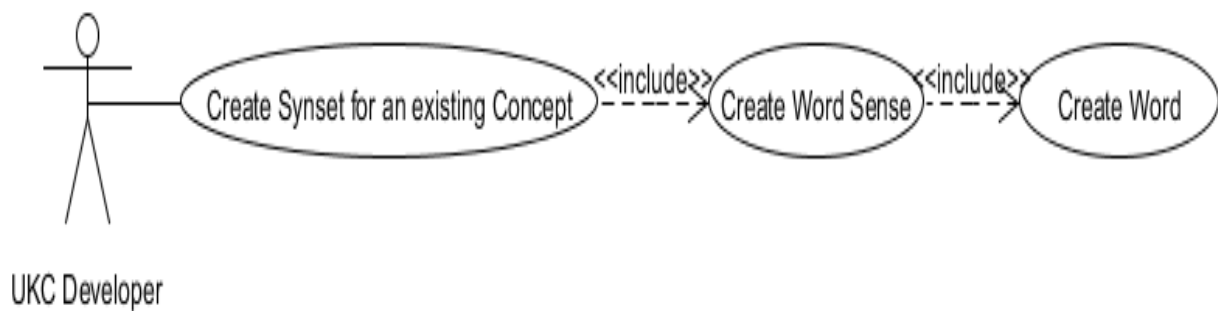
3- User fills the interactive form with the new synset details. The form requires the following mandatory fields:

- Set of word senses.
- Part of Speech (POS).
- Gloss
- Example(s)

4- User saves the changes.

5- System checks if the new synset includes new word senses. . If it contains any new word sense, system creates the new word sense and stores the word in the linguistic resource.

6- System creates a new synset and associates it with the language-independent concept



6- Update Synset

Main Scenario: Update synset

1- User retrieves the synsets of any natural language word by following the scenario of use case: [Read Synsets in the Working Language](#) or by reading the lexicalization of a language-independent concept by following the scenario of use case:

[Read the Lexicalization of a Concept](#).

2- User selects to update the retrieved synset

3- System displays an interactive editable form for editing the synset. The same interactive form defined for creating a new synset.

4- User may choose to update the POS, gloss, or example sentences.

Sub Scenario-1.1 : Create new Word Sense

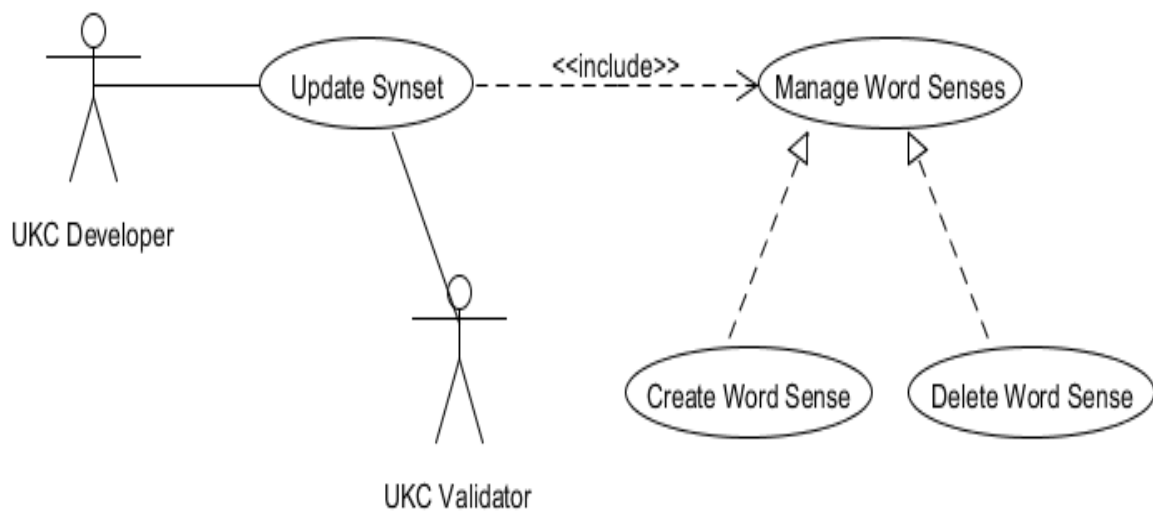
5- User selects to create a new word sense. user can create a new sense by creating a new word with its exceptional forms if any.

Sub Scenario-1.2 : Delete Word Sense

5- User selects to delete one of the defined word senses for the synset.

6- User saves the changes.

7- System validates the updates and save the changes.



7- Delete Synset

Main Scenario: Delete Synset

1- User retrieves the synsets of any natural language word by following the scenario of use case [Read Synsets in the Working Language](#). or by reading the lexicalization of a language-independent concept by following the scenario of use case [Read the Lexicalization of a Concept](#).

2- User selects to delete the retrieved synset.

3- System displays a confirmation dialog asking the user to confirm the deletion.

4- User confirms the deletion.

Sub Scenario-1: Delete Word Senses

5- System checks each word sense, if the word sense is not part of any other synset, system deletes the word sense.

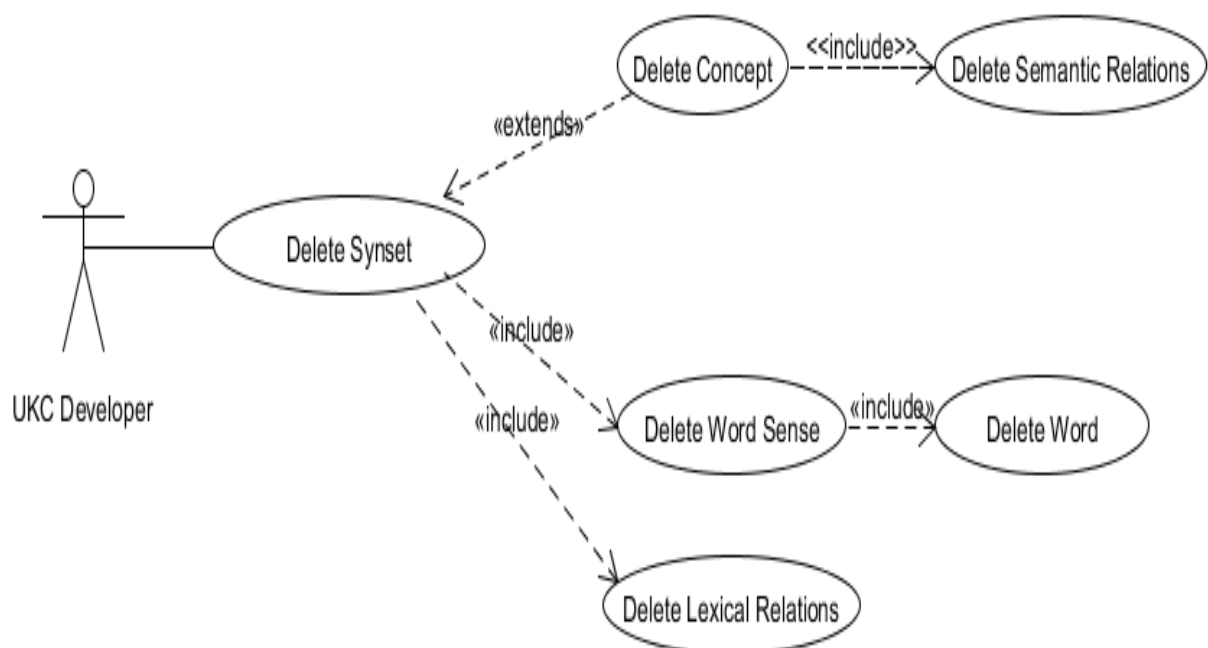
Sub Scenario-2: Delete Lexical Relations

6- System checks if the synset has lexical or semantic lexical relations with other synsets, if relations exist, system deletes them.

Extension Scenario-1: Delete Concept

7- System checks if there is any other lexical gaps still associated with the same concept. If the system finds a lexical gap associated with the same concept, the delete operation will be refused and the system will show an error message informing the user that the existing lexical gap must be deleted first. Knowing that, concepts can't be associated with only lexical gaps, it must be associated with at least one language-dependent synset.

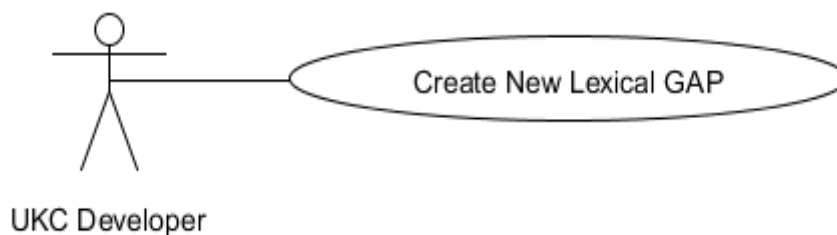
8- If the concept associated with this synset is not associated with any *synset* or any lexical gap in another natural language, the system deletes the associated concept as well.



8- Create New Lexical GAP for an existing Concept

Main Scenario: Create New Lexical GAP for an existing concept

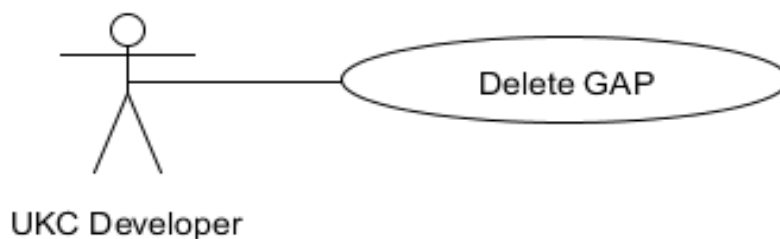
- 1- User selects a concept with no lexicalization in the working language.
- 2- User selects to define a new lexical GAP in the working language for the retrieved concept.
- 3- System defines a new lexical GAP and associates it with the language-independent concept



9- Delete Lexical GAP

Main Scenario: Delete lexical gap

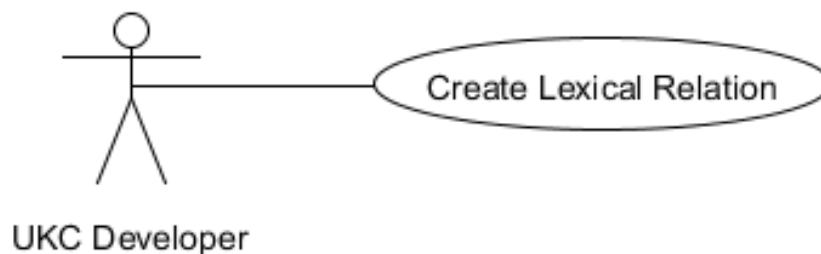
- 1- User selects a lexical gap in the working language.
- 2- User selects to delete the retrieved lexical gap.
- 3- System displays a confirmation dialog asking the user to confirm the deletion.
- 4- User confirms the deletion.
- 5- System deletes the lexical gap from the linguistic resource.



10- Create Lexical Relation

Main Scenario: Create Lexical Relation

- 1- User retrieves the synsets of any natural language word by following the scenario of use case: [Read Synsets in the Working Language](#). or by reading the lexicalization of a language-independent concept by following the scenario of use case: [Read the Lexicalization of a Concept](#).
- 2- User selects to define new lexical relation for the retrieved synset.
- 3- System displays an interactive form containing the following fields: source synset, target synset, and the set of possible relations to choose one from. The source synset is pre-entered by the System and can't be updated.
- 5- The user enters the relation and the target synset.
- 6- User saves the changes.
- 7- System validates the constraints for creating lexical relations, if the input satisfies the constraints, system creates the new lexical relation.



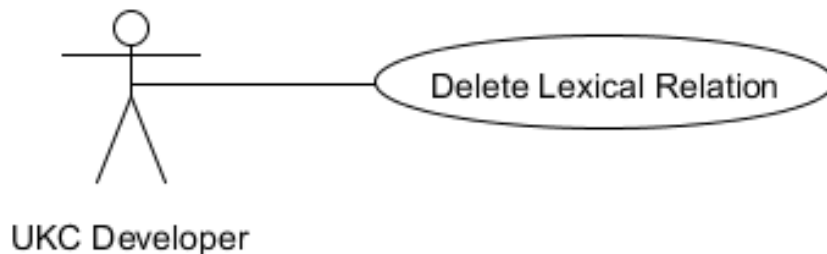
11- Delete Lexical Relation

Main Scenario: Delete Lexical relation

- 1- User retrieves the synsets of any natural language word by following the scenario of use case [Read Synsets in the Working Language](#) or by reading the lexicalization of a language-independent concept by following the scenario of use case [Read the Lexicalization of a Concept](#).
- 2- User retrieves the lexical relations for the retrieved synset.
- 3- User selects to delete one of the relations.
- 5- System displays a confirmation dialog asking the user to confirm the deletion.

6- User confirms the deletion.

7- System validates the constraints for deleting lexical relations, if the input satisfies the constraints, system deletes the lexical relation.



12- Create Semantic Relation

Main Scenario: Create Semantic Relation

1- User retrieves any language-independent concept by following the scenario of use case [Read the Concept of a Synset](#).

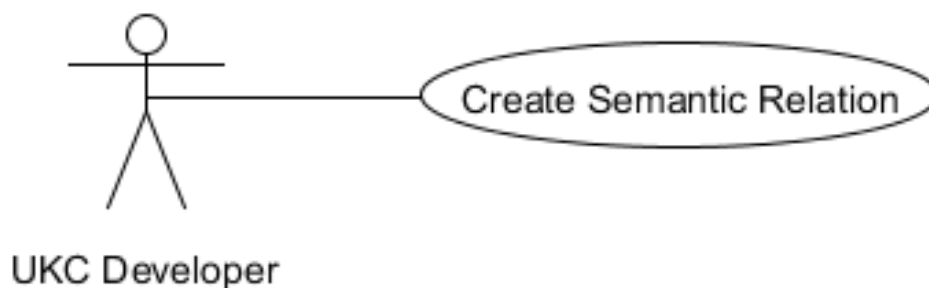
2- User selects to define new semantic relation for the retrieved concept.

3- System displays an interactive form containing the following fields: source concept, target concept, and the set of possible semantic relations. The source concept is pre-entered by the System and can't be updated.

5- The user selects the semantic relation and the target concept.

6- User updates the interactive form and save the changes.

7- System validates the constraints for creating new hierarchical and associative semantic relations. If the input satisfies the constraints, system creates the new semantic relation.

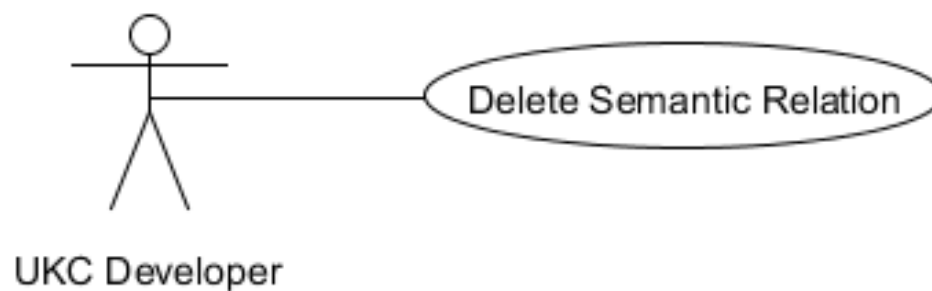


13- Delete Semantic Relation

Main Scenario: Delete Semantic Relation

- 1- User retrieves any language-independent concept by following the scenario of use case [Read the Concept of a Synset](#).
- 2- User retrieves the semantic relations for the retrieved concept.
- 3- User selects to delete one of the relations.
- 5- System displays a confirmation dialog asking the user to confirm the deletion.
- 6- User confirms deletion.

- 7- System validates the constraints for deleting hierarchical and associative semantic relations. If the input satisfies the constraints, system deletes the semantic relation.



OVERALL ARCHITECTURE

The overall architecture is a conceptual model that represents the main system components and data transfer between them. The overall architecture (Figure 6.2) was designed as a multi-layer (Model-View-Controller) architecture, commonly known as MVC pattern, a software design pattern for implementing user interfaces. The MVC architecture is divided into three main interconnected components in order to separate the application information and business logic from the ways that information is presented to the user.

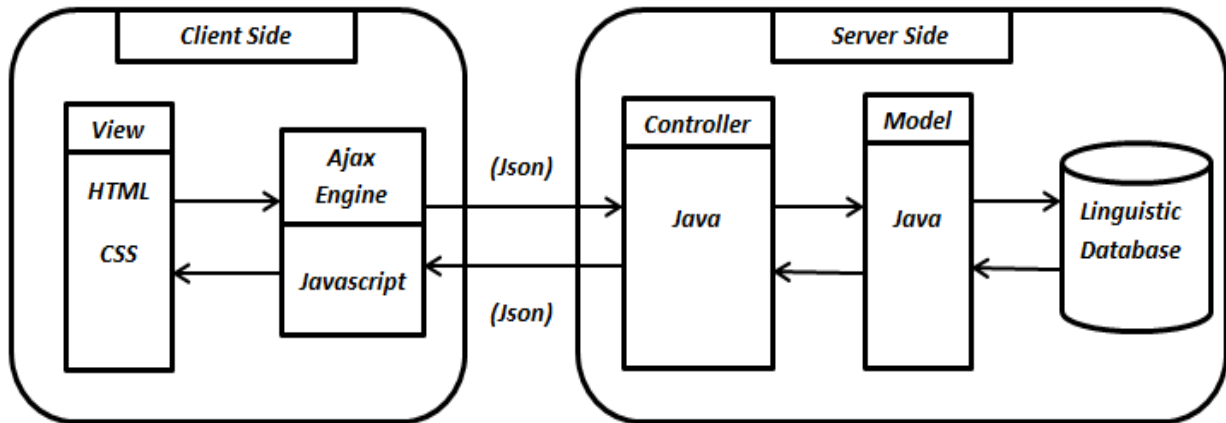


Figure 6.2- UKC Interactive Platform Overall System Architecture

In Figure 6.2, the client side is the user's browser environment. We have implemented two main components that run on the client side; View Component and Ajax Engine. The View component is responsible for data representation. The technologies used in data representation are: Hypertext Markup Language (HTML), the standard language for creating web pages, and Cascading Style Sheets (CSS), a style sheet language for formatting web pages. On the other hand, the Ajax Engine is the core of the web application which implements the application logic using JavaScript as a web programming language and JavaScript Object Notation (JSON) as a lightweight data-exchange format between the client and server. The Ajax Engine runs within the user's browser to ensure prompt responses to the user requests. The added engine eliminates the 'click and wait' nature of the classic web applications and responds instantly to the user actions by exchanging data with the server behind the scenes without refreshing the web page.

In Figure 6.2, the server side is composed of two main components; the web controller and the application data mode, both components are using Java as a programming language. The Java controller is responsible for handling communication with the client side through the Ajax Engine and submitting commands to the application data model for reading or updating application data. On the other hand, the application data model is the central location for application core data, business logic, and functions accessing the linguistic database.

USER INTERFACE DESIGN

Figure 6.3 shows the application user interface. The user interface is divided into 5 main regions: (1) search panel (2) natural language core panel, named synsets panel (3) concepts core panel, named concepts panel, (4) expandable drawer used as linguistic reference panel, and (5) color legend at the bottom to differentiate between working and reference languages.

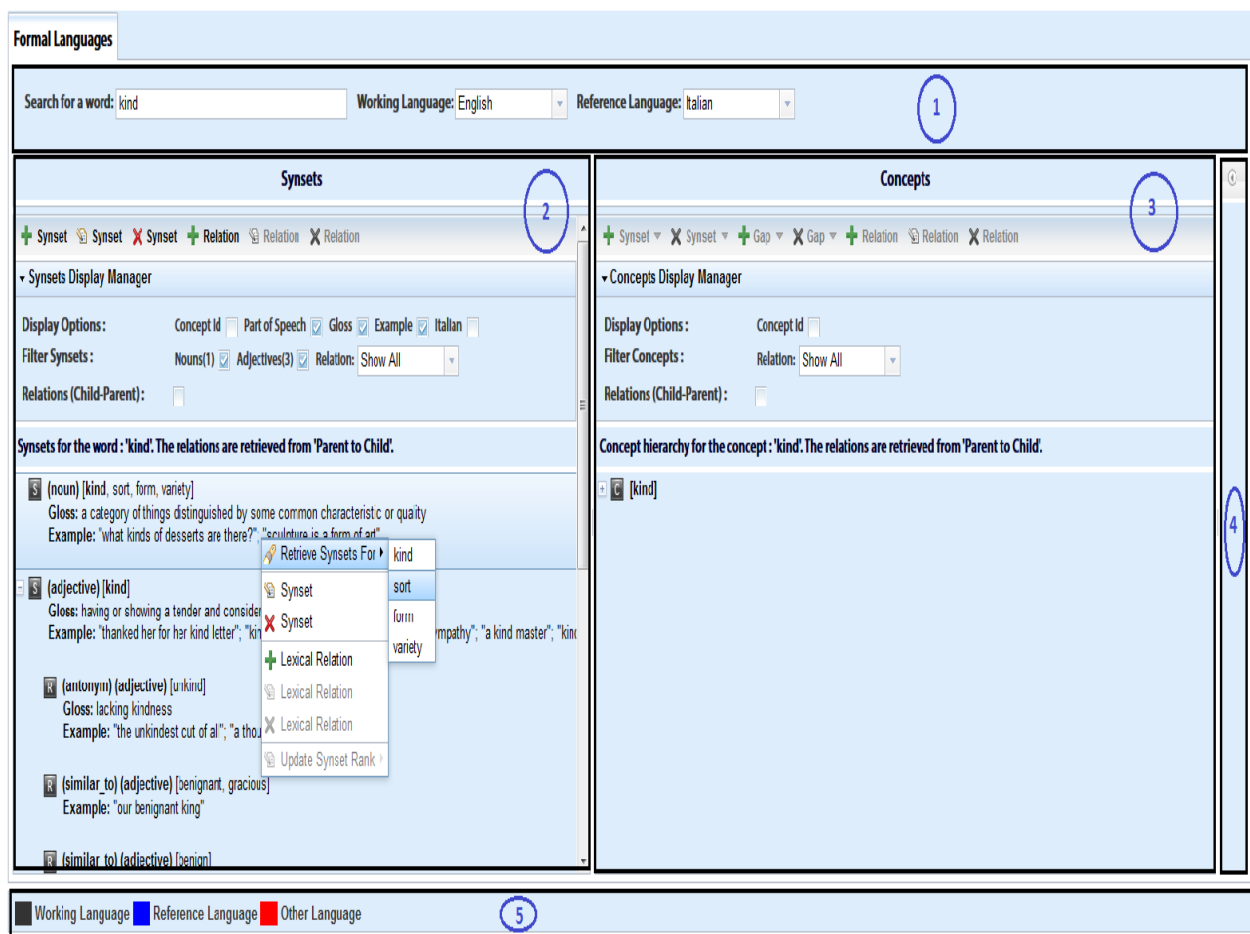


Figure 6.3 UKC Interactive Platform User Interface

The top region where the user can start a new search by typing a word and choosing the desired working and reference languages respectively. The working language is the default language, when the user performs a search or an update operation; the system applies the changes based on the selected working language. The reference language is

mainly for multilingual support in order to view the working language synset in another language or a lexical gap if there is no corresponding synset. The set of color legends at the bottom of the screen is used to differentiate between working language synsets (black font), reference language synsets (blue font). Another possible case when the language independent concept label is obtained from another language different from both working and reference languages. In this case, the concept label will be also retrieved and highlighted as a label from another language (red font).

The synsets panel displays the content retrieved from the natural language core; the word synsets and their lexical and semantic lexical relations. The user can interact with the natural language core contents through the following user interface components:

- Toolbar at the top for manipulating the displayed synsets by performing Create-Update-Delete operations on synsets and their relations.
- Display manager for updating the visual display of the displayed synsets. For each synset, it's possible to show or hide the: synset gloss, example sentences, the language-independent concept identifier, or the corresponding synset in the reference language.
- Synset filtration capability for the displayed synsets, it's possible to filter synsets by part of speech or type of lexical relation.
- Viewing lexical and semantic relations from parent to child (sub-relations) or from child to parent (super-relations).
- Contextual (right-click) menu for providing handy way to retrieve synsets for synonym words and performing create/update/delete operations on synsets.

The concepts panel displays the content retrieved from the concept core; the language-independent concepts and their semantic relations. The user can interact with the concept core contents through the following user interface components:

- Toolbar at the top for defining new synset or lexical gap in the working or the reference language for the selected concept.

- Display manager for updating the visual display of the displayed concepts. For each concept, it's possible to show or hide the language-independent concept identifier.
- Concept filtration capability for the displayed concepts, it's possible to filter concepts by relation type.
- Viewing semantic-lexical relations from parent to child (sub-relations) or from child to parent (super-relations).
- Contextual (right-click) menu for providing handy way to create or delete synsets and lexical gaps and performing create/update/delete operations on semantic relations between concepts

A dynamic synchronization between the synsets panel and concepts panel regions takes place when the user selects any synset from the left region, the system automatically display the corresponding concept in the right region.

CRUD operations for the UKC linguistic items operations are performed through the use of interactive and well-designed input forms, For instance (Figure 6.4) shows the new synset input form. The interactive form allows the user to do the following:

- Define new concept Id and label for the new concept or choose an already existing concept Id.
- Manage word senses by adding, removing, and defining the order (word sense rank) of any number of word senses.
- Define a gloss in any language using a virtual keyboard.
- Define an example in any language using a virtual keyboard.
- Choose the POS (noun, verb, adjective, or verb)
- Upload a file, an optional file that the linguistic expert may upload as a claim file referring to other linguistic resources or any explanation related to the synset definition.

Create Synset

Concept

Concept Id :

Concept Label :

Word Senses

Words :

+ Add Word - Remove Word

Synset Details

Gloss :

Example :

Part of Speech:

File Upload

Upload File:

#	Type	File Name	Size

Figure 6.4 Interactive input form for the creation of a new synset.

The expandable drawer on the right is used as a (read-only) linguistics reference panel. The purpose of this panel is to assist users while working on the main synsets and concepts panels in retrieving other synsets or concepts and use them as a reference without the need to erase the contents of the main panels.

The reference panel also assists user in creating new lexical and semantic relations through a drag and drop facility. For instance (Figure 6.5), a new lexical relation can be created by dragging a reference synset from the reference panel and dropping it over another synset from the main synsets panel.

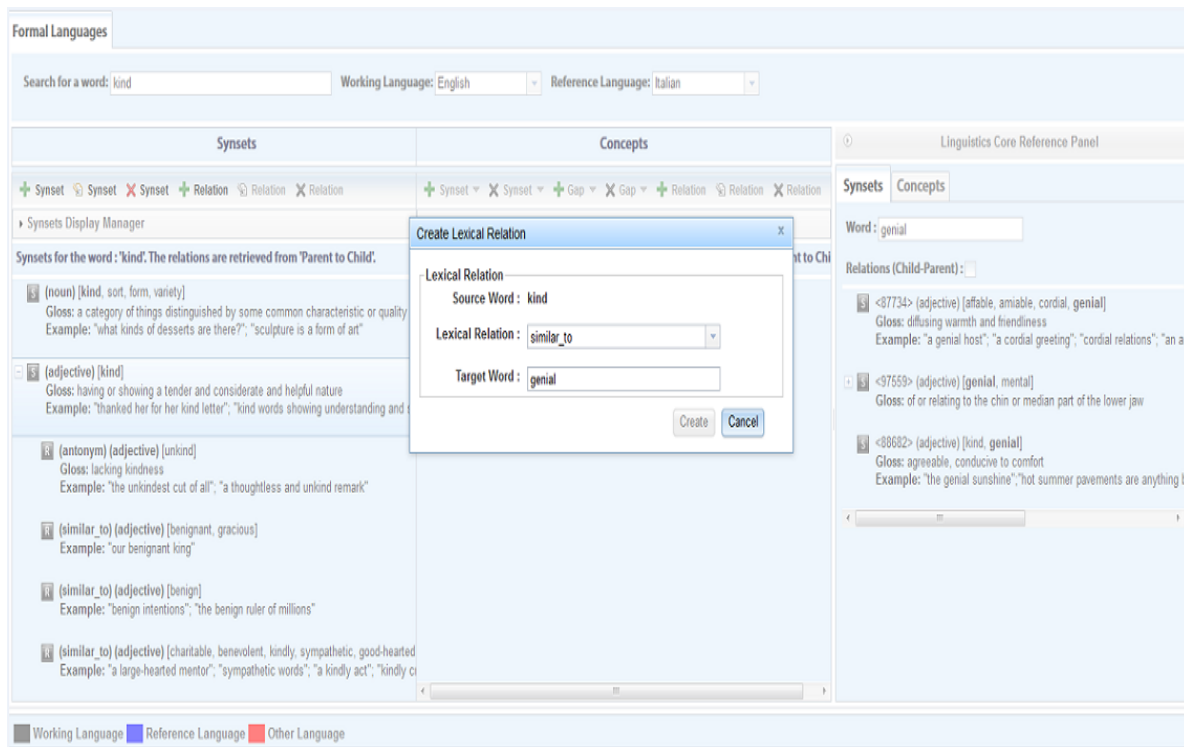


Figure 6.5 Creation of new Lexical Relation using drag and drop.

Using the main and reference panels accompanied by toolbars and contextual menus together with the provided full control over the displayed information through the display managers, we should end up having an elegant linguistic analysis and manipulation tool which allow linguistic experts to enrich the available linguistic resources with minimal effort.

Chapter 7

COMPARATIVE EVALUATION (WORDNET VS. UKC)

In order to improve the usability of the UKC web platform and knowing that the WordNet application is considered as the de-facto standard, we started by performing a comparative analysis between the two interfaces in order to elicit the usability problems on the UKC platform.

By being a comparative evaluation, it should be kept in mind that the two systems should share some meanings, some operation. In this case, even if the UKC view has been designed as a full Create-Retrieve-Update-Delete (CRUD), while the WordNet view is a simple knowledge retrieval interface. We took this into account, the designed tasks for the evaluation, like the rest of the work, were built only around the retrieval features of the two interfaces.

QUANTITATIVE WORK

On the base of these assumptions, and following the directives of Bodker [Bodker, S. (2000)], a quantitative analysis has been conducted to investigate how, in general, the user react to both views. For the quantitative work, the key points were three: tasks, questionnaires and measures. The main tasks that have been proposed to the testers were the following:

- Perform a search for an arbitrary word in an arbitrary language.
- Search for an arbitrary lemma in multiple idioms.
- Filter the results of a search process on the base of their Part-Of-Speech.
- Retrieve semantic relations insisting on a given concept.

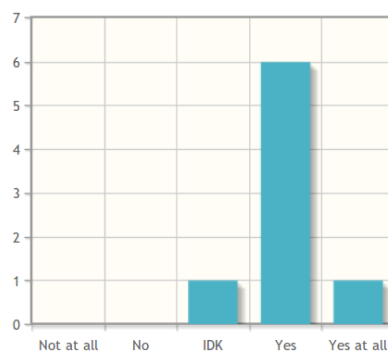
The entire process of comparative analysis was supported by a custom-made platform capable of switching the two views shown to the user in order to allow for performing each task on both of them. Along with this capability, the platform was entitled to keep

track of the time spent to perform each task, to submit questionnaires, after each group of tasks, and collect their results. The time spent and the results of the questionnaire were used to perform analysis and plot graphs. This tool was tested with two pilot runs. The two testers involved in this case were a developer and a PhD in Human Computer Interaction, which gave important suggestions about the usability of the platform itself.

EVALUATION RESULTS

In this paragraph are reported the results of the comparative evaluation. Before proceeding with the results it is important to state the nature of the test subjects. They were directly contacted among the members of our research group. The selection criteria was based on the fact that they were almost all users of the WordNet interface, thus they can be considered domain experts. The number of involved test subjects was 8, of those, 5 were PhD, 2 post-doc and a software engineer. Their answers in the form of Yes, No, or I don't know (IDK) were analyzed and plotted as bar graphs.

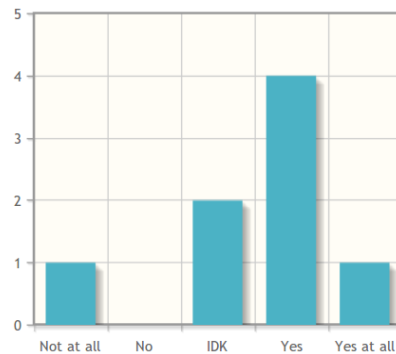
- **Results display:** the WordNet way of displaying results has been judged not easy to read and interpret. In particular, users have preferred the UKC way of displaying results in two panels (Synsets and Concepts). In this way, each synset is easily identifiable.



UKC Panels View, is it intuitive?

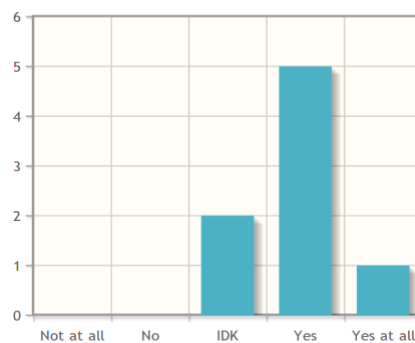
- **Results aggregation:** with respect to the UKC, the WordNet view performs results aggregation over the data. In particular results are grouped according to their part of speech tag (POS). The UKC view includes all the results in the same

table, thus the only way the user have to identify their type is to look at the POS-tag field or filter by POS or relation type. Indeed, more space that can be saved.



UKC Display Manager, is it useful?

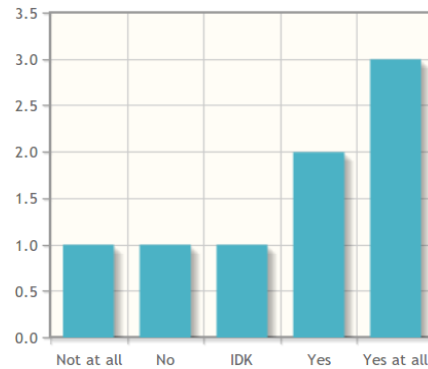
- **Results relations:** another important point involved in the analysis is the relation retrieval feature. By being a linguistic resource, the interface should provide the user access to the relations insisting on a concept in a way to easy the navigation among them. Under this point of view, the users have preferred the way the UKC interface manages relations retrieval and displaying. In particular the capability of filtering and interact with these relations, with respect to the fixed and link-based of WordNet, has been proven to be more flexible to the user needs.



Is UKC relation retrieval approach more complete?

- **Multilingual Results:** the UKC has been proven to be more productive when it comes to mapping results in different languages with respect to WordNet. In particular, the former is able to retrieve the mapping of a results in each selected

language, while the latter allows the user to retrieve results just in one language, namely losing the mapping implicit function.



Multilingual WordNet, is it difficult?

- **Missing history:** beside these quantitative results, another important point got from some qualitative interviews, is the fact that both the interfaces do not keep track of the user's history of actions. Even though there was no task about it, users have reported this missing feature.

Chapter 8

UKC COLLABORATIVE PLATFORM

In this chapter, we introduce the collaborative platform for UKC development that was designed and implemented based on our methodological approach for collaborative development using collaborative objects and collaborative workflows (chapter 4).

The collaborative development process takes place in two main phases: *development phase* and *validation phase*. The conditions, constraints, transitions between the two phases are defined using the collaborative workflows.

In this chapter, we explain the user studies, requirement specifications together with the application overall architecture and user interface design.

USER STUDIES

We performed specific user studies for the collaborative platform by conducting interviews with knowledge experts. The user studies revealed that a new category of user, *UKC validators*, is required. In addition, two new roles have been assigned to the UKC Administrator which are: (1) assigning collaborative workflows to collaborative objects; and (2) creation of interactive polls.

The revised categories of users for the collaborative platform become:

- **UKC Users:** those who are allowed to view the UKC knowledge base for the navigation and linguistic analysis without performing any updates to the linguistic resource.
- **UKC developers:** knowledge experts and domain developers who are able to access the Read/Write interactive feature and perform development and maintenance operations on the UKC knowledge base. The tasks that could be performed by UKC developers include: (1) Linguistic resource enrichment, (2)

Linguistic resource translation, (3) ontology development, and (4) ontology localization.

- **UKC validators:** top level expert who can approve the proposals and the changes made by UKC developers. UKC validators are allowed to access the Read/Write interactive features and UKC approval services.
- **UKC Administrators:** category of users have full access to system features and they can perform administrative tasks such as creating users, creating groups and assigning user roles to users, assigning collaborative workflows to collaborative objects and creation of interactive polls. Administrators are responsible for applying the collaborative development methodology beside facilitating the communication between the participants.

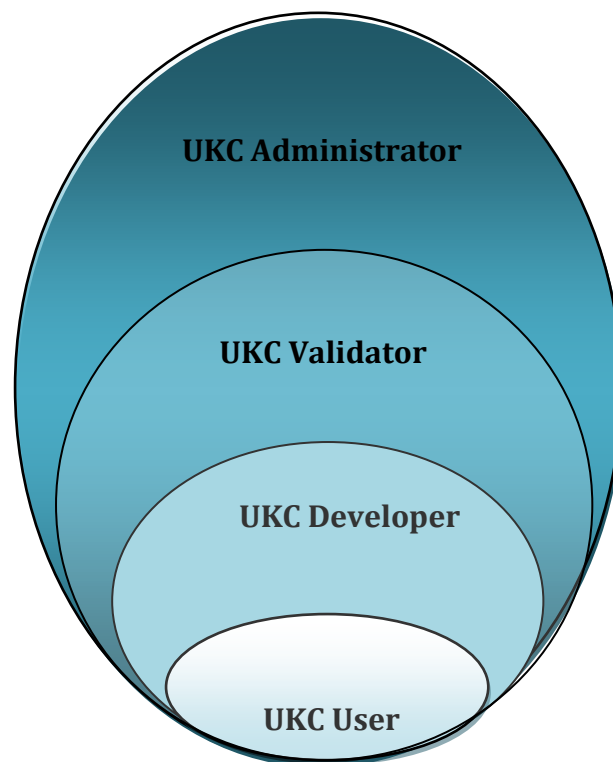


Figure 8.1- Access rights for different categories of users; End Users have read only privileges, UKC Developers have read/write privileges; UKC validators have read/write and approval privileges and UKC Administrators have full access.

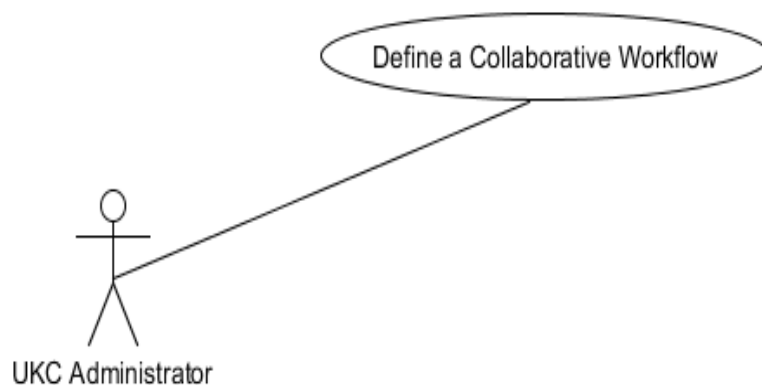
REQUIREMENT SPECIFICATIONS

In the following subsections, we explain the requirement specifications for supporting the collaborative development environment. This section is considered as an extension to the requirements specifications defined in section 6.2 for manipulating UKC linguistic items; synstes, lexical gaps, concepts, lexical and semantic relations.

1- Define a Collaborative Workflow

Main Scenario: Define a Collaborative Workflow

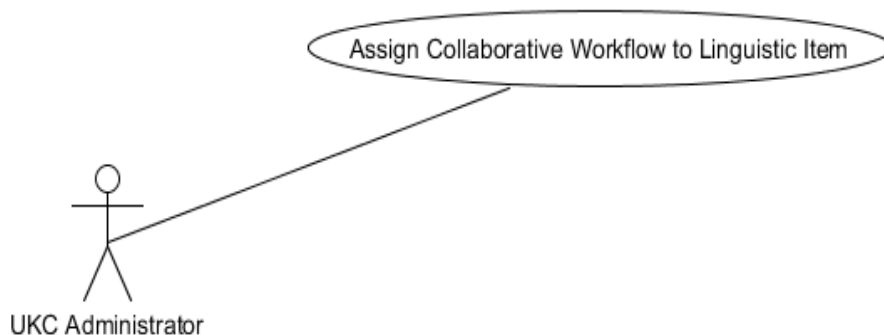
- 1- UKC Administrator chooses to define a new collaborative workflow.
- 2- System lists the set of available user roles (UKC Developer and UKC Validator) with the possibility to define new user roles and the set of available collaborative workflow nodes (state, tasks, condition, notification, timer, and the rest of the nodes), knowing that the complete list of process nodes is defined in chapter 4.
- 3- UKC Administrator defines the new collaborative workflow by defining the details, conditions, and constraints for the UKC development and validation tasks.
- 4- UKC Administrator saves the changes.
- 5- System stores the new collaborative workflow.



2- Assign Collaborative Workflow to Linguistic Item

Main Scenario: Assign Collaborative Workflow to Linguistic Item

- 1- UKC Administrator chooses to view the list of defined collaborative objects and the list of defined collaborative workflows.
- 2- System displays the set of defined UKC collaborative objects (Synset, Lexical Gap, Lexical Relation, and Semantic Relation), knowing that the concept object is auto-generated therefore it's not a collaborative object.
- 3- System lists in the same view the defined collaborative workflows (single-user approval, multiple-user approval, etc.)
- 4- UKC Administrator chooses to assign one of the collaborative objects to one of the collaborative workflows.
- 4- UKC Administrator saves the changes.
- 5- System stores the new configuration.



3- Manipulate Linguistic Item Collaboratively

Main Scenario: Manipulate Linguistic Item

- 1- UKC developer manipulates one of the linguistic items by performing one of the CRUD operations.
- 2- Workflow engine assigns the task to one or more UKC Validators according to the defined collaborative workflow.

Extension Collaborative Scenario: Approve Linguistic Item

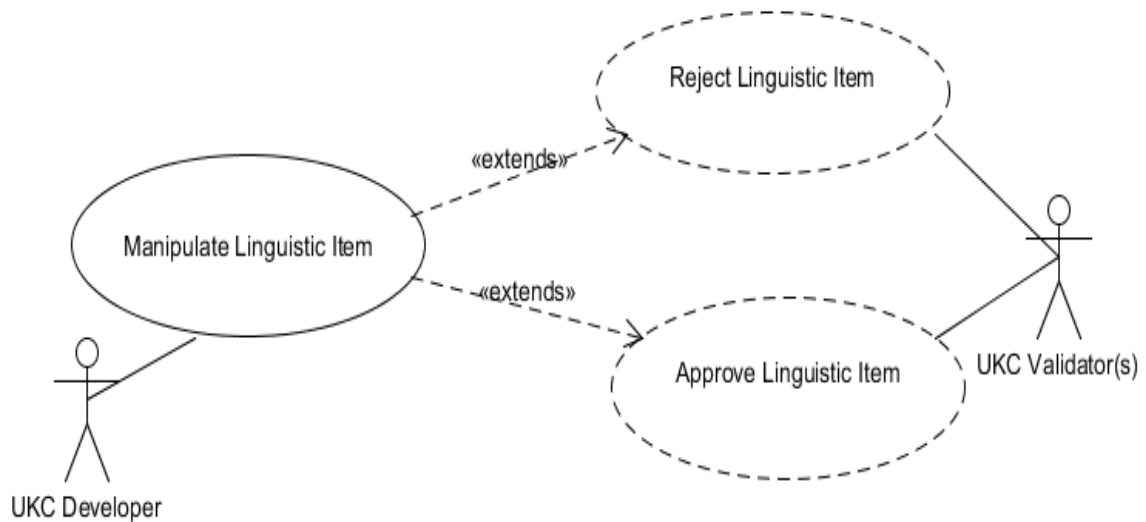
4- UKC Validator(s) check in collaboration the linguistic item manipulated by the UKC developer in collaboration and decide to approve it.

5- Workflow engine checks that the conditions for approving the linguistic item haven been satisfied and marks the item as approved.

Extension Collaborative Scenario: Reject Linguistic Item

4- UKC Validator(s) check in collaboration the linguistic item manipulated by the UKC developer and decide to reject it.

5- Workflow engine assigns the task back to the UKC developer together with the comments provided by the Validators.



4- View Developed Items List

Main Scenario: View Developed Item List

1- UKC developer chooses to view his/her list of developed linguistic items.

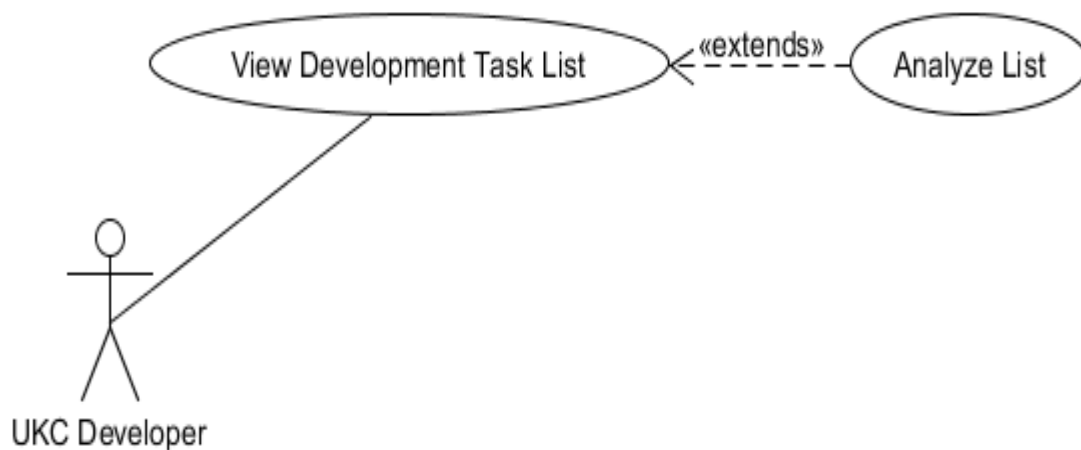
2- System displays the list of developed linguistic items by the UKC developer. System shows the following details:

- Development date
- Linguistic Item (Synset, Lexical GAP, Lexical Relation, or Semantic Relation)
- Manipulation Type (Create, Update, or Delete)

- Development is “Completed” or still “Pending approval”.
- Comments and discussions associated with the developed item.

Extension Scenario-1: Analyze List

3- UKC developer may decide to analyze the displayed list of items by filtering them by either “Completed Tasks” or “Pending Approval Tasks”. Another possibility is sorting by date or semantic relations between developed concepts.



5- View Validated Items List

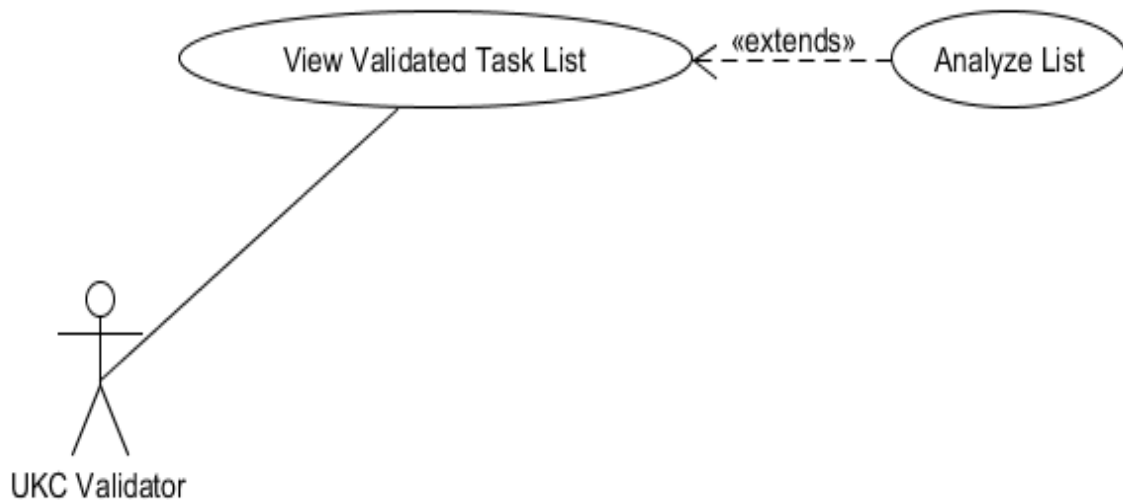
Main Scenario: View Validated Item List

1- UKC Validator chooses to view his/her list of validated linguistic items.
 2- System displays the list of validated linguistic items by the UKC Validator. System shows the following details:

- Development date
- Validation date
- Linguistic Item (Synset, Lexical GAP, Lexical Relation, or Semantic Relation)
- Manipulation Type (Create, Update, or Delete)
- Validation is either “Completed” or still “Pending Approval”.
- Comments and discussions associated with the developed item.

Extension Scenario-1: Analyze List

3- UKC Validator may decide to analyze the displayed list of items by filtering them by either “Completed Tasks” or “Pending Approval Tasks”. Another possibility is sorting by date or semantic relations between developed concepts.



6- Initiate Discussion

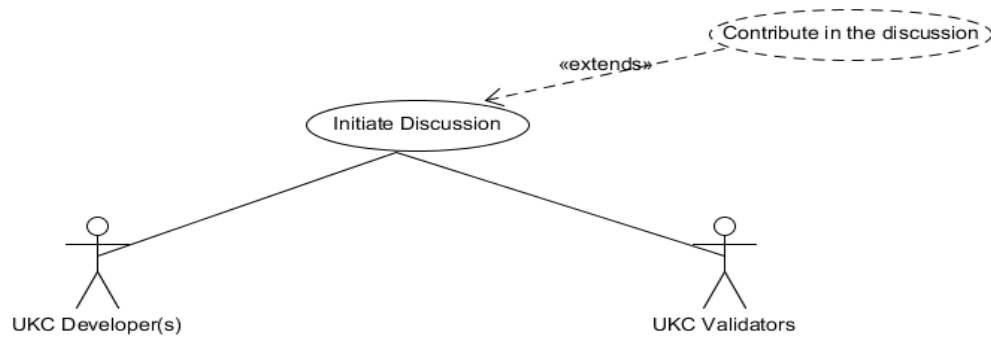
Main Scenario: Initiate Discussion

- 1- User chooses to initiate a new discussion in a specific topic or about the development of a specific linguistic item.
- 2- System initiates a new discussion thread in the desired topic. The system should allow for the following

- Organize discussions in categories.
- Sort discussions by category or date.

Extension Collaborative Scenario: Contribute in the discussion

- 3- User(s) participate in the discussion and post their contributions regularly until they reach a common agreement.



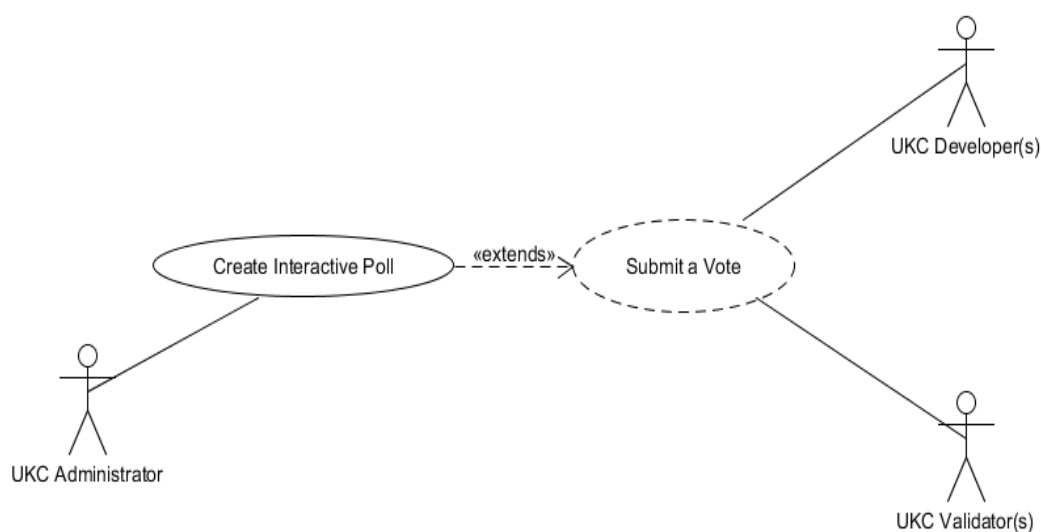
7- Create Interactive Poll

Main Scenario: Create Interactive Poll

- 1- UKC Administrator chooses create new interactive poll. The poll is in the form of multiple choice type question and answers.
- 2- System publishes the new interactive poll for voting.

Extension Collaborative Scenario: Submit a Vote

- 3- User(s) participate in the interactive poll by submitting their votes. Each user either UKC developer or Validator is allowed to vote once.
- 4- System counts the votes and displays the result in percentage.



OVERALL ARCHITECTURE

UKC collaborative platform is designed and implemented as a *portlet application*. In order to understand the portlet application architecture, we start by defining a *portlet* as a self-contained web application, i.e. mini web application that has its own web pages, services and data sources. A *web page* containing a group of portlets integrated together in a consistent and systematic way is called a *portal page*. Therefore, a web portal could be defined as web page that brings multiple web applications or portlets together and allows for effective communication and integration between them.

Portlet applications are managed by a *portlet container* which provides the environment for portlet management and forms the infrastructure required for running a portlet application. A portlet container's responsibilities include managing portlet instances and handling communication between portlets and with data sources. There are several open-source portlet containers available now days. The choice of a portlet container plays an important role in portlet application projects, since it can help reduce the development time by providing built-in portlets and the ability to access container's built-in portlets features from the newly custom portlets. The choice of a suitable portlet container should be driven by the project requirement specifications. We have conducted a comparative study of the available open-source portlet containers and decided to use *Liferay*⁹. Liferay is an open source portlet container that comes with built-in portlets for web 2.0 social and collaboration features. In addition, it has built-in workflow engine that allows for running custom defined workflows. Liferay provides a robust platform for building social and collaborative portlets that could be extended and customized according to any project requirement specifications which perfectly fits with our methodology and requirements.

Figure 8.2 shows the overall architecture for the UKC collaborative platform. It's composed of Liferay portelt container hosting four portlets: (1) UKC portlet (2) Administration Portlet (3) Discussion Portlet; and (4) Polls Portlet and two data sources: (1) UKC linguistic database and (2) Information management database.

⁹ www.liferay.com

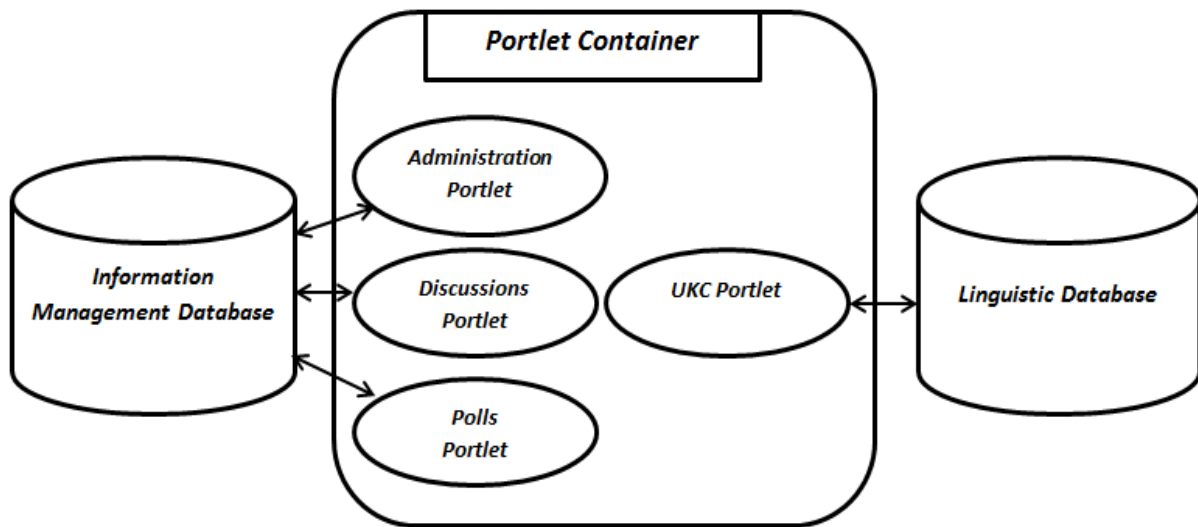


Figure-8.2 UKC Collaborative Platform Overall Architecture

The responsibilities are shared among the four portlets and the portlet container which ensures smooth data exchange between the portlets.

- **UKC Portlet:** is responsible for manipulating linguistic items and management of developed and validated lists of linguistic items. This portlet accesses the linguistic data source.
- **Administration Portlet:** is responsible for providing administration services such as user management, collaborative objects definition, workflows definition and assigning collaborative objects to collaborative workflows. The administrative information is stored in the information management database.
- **Discussions Portlet** is responsible for handling the discussion boards, creating new discussion threads and management of ongoing discussions. The discussions are stored in the information management database.
- **Polls Portlet** is responsible for handling polls, creating polls, displaying polls and counting poll results. The polls discussions are stored in the information management database.

USER INTERFACE DESIGN

Figure 8.3 shows the UKC collaborative platform user interface. The main interface for developers and validators is composed of five tabs: (1) The Home page (2) UKC Portlet, (3) Task Notifications list of developed and validated linguistic items. (4) Discussion boards; and (5) Interactive Polls.

Figure 8.3 shows the UKC portlet tab selected; the portlet interface is the same interface used for the UKC Interactive platform (chapter 6) and provides the same functionalities for UKC task development. In addition, the interface provides the linguistic provenance details visualized in orange font, i.e. the source of linguistic item which could be the name of one of the UKC developers or another linguistic resource like WordNet or MultiWordNet where the item has been imported.

The screenshot displays the UKC Collaborative Platform user interface. At the top, there is a navigation bar with tabs: Home, UKC, Task Notifications, Discussion Boards, and Polls. The UKC tab is selected. Below the navigation bar, there is a search area with a search box containing the word 'kind', a Working Language dropdown set to 'English', and a Reference Language dropdown set to 'Italian'. The main content area is divided into two columns: Synsets and Concepts. The Synsets column shows a list of synsets for the word 'kind', including a noun and an adjective. Each synset entry includes a gloss, an example, and a source. The Concepts column shows a concept hierarchy for the concept 'kind'. At the bottom of the interface, there is a legend for Working Language (black), Reference Language (blue), and Other Language (red).

Figure 8.3 UKC Collaborative Platform User Interface

Figure 8.4 shows the Task Notification tab selected; the user whether UKC Developer or UKC Validator is allowed to view the list of tasks and filter tasks by state (Pending or Completed). It also allows sorting the task list by date or the relation between the concepts and the parent concept.

Two navigation options are provided “View Task” and “View in UKC”. The former allows viewing all the linguistic item details and any associated comments and discussions and the later allows for viewing the linguistic item in the UKC portlet interface in order to visualize the item in context or perform more linguistic analysis operations.

The screenshot displays the UKC (Universal Knowledge Core) interface. At the top left is the UKC logo. A navigation bar contains 'Home', 'UKC', 'Task Notifications', 'Discussion Boards', and 'Polls'. Below this, there are filter options: 'Display:' with 'Pending Tasks' and 'Completed Tasks' (both checked), and 'Sort By:' with a dropdown menu set to 'By Date'. A 'LINGUISTIC ITEM LIST' button is visible in the top right. The main content is a table with the following data:

Date	Notification Message	Linguistic Item	Linguistic Operation	Task Type	Task Completed	View Task Details	View in UKC
Sep 26, 2014, 8:41:21 PM	Ahmed Tawfik sent you a linguistic item for review.	Synset	Update	review	No	View Task	View in UKC
Sep 26, 2014, 8:39:08 PM	Your submission was rejected by Ahmed Tawfik, please modify then resubmit.	Synset	Update	update	Yes	View Task	View in UKC
Sep 26, 2014, 8:38:16 PM	Ahmed Tawfik sent you a linguistic item for review.	Synset	Update	review	Yes	View Task	View in UKC
Sep 26, 2014, 8:37:29 PM	Ahmed Tawfik sent you a linguistic item for review.	Synset	Update	review	Yes	View Task	View in UKC
Sep 26, 2014, 8:34:03 PM	Ahmed Tawfik sent you a linguistic item for review.	Synset	Delete	review	No	View Task	View in UKC
Sep 6, 2014, 12:35:59 AM	Ahmed Tawfik sent you a linguistic item for review.	Synset	Delete	review	No	View Task	View in UKC
Sep 6, 2014, 12:35:08 AM	Ahmed Tawfik sent you a linguistic item for review.	Synset	Delete	review	No	View Task	View in UKC
Sep 3, 2014, 7:08:40 PM	Israt Jahan sent you a linguistic item for review.	Synset	Create	review	No	View Task	View in UKC
Sep 3, 2014, 6:58:04 PM	Israt Jahan sent you a linguistic item for review.	Synset	Create	review	Yes	View Task	View in UKC

Figure 8.4 Task Notification List

The tasks listed in the “Task Notifications” are either development tasks or validation tasks. Development tasks are performed through the UKC portal interface. On the other hand, Validation tasks are performed through a specific task validation page accessed by the UKC Validators (Figure 8.5).

Figure 8.5 shows the Task validation page, the page presents clearly the assignment of the linguistic item under development to which UKC Validator, the current state of the task, creation and due dates for the task and the possibility to approve, reject, or assign the task to another UKC Validator.

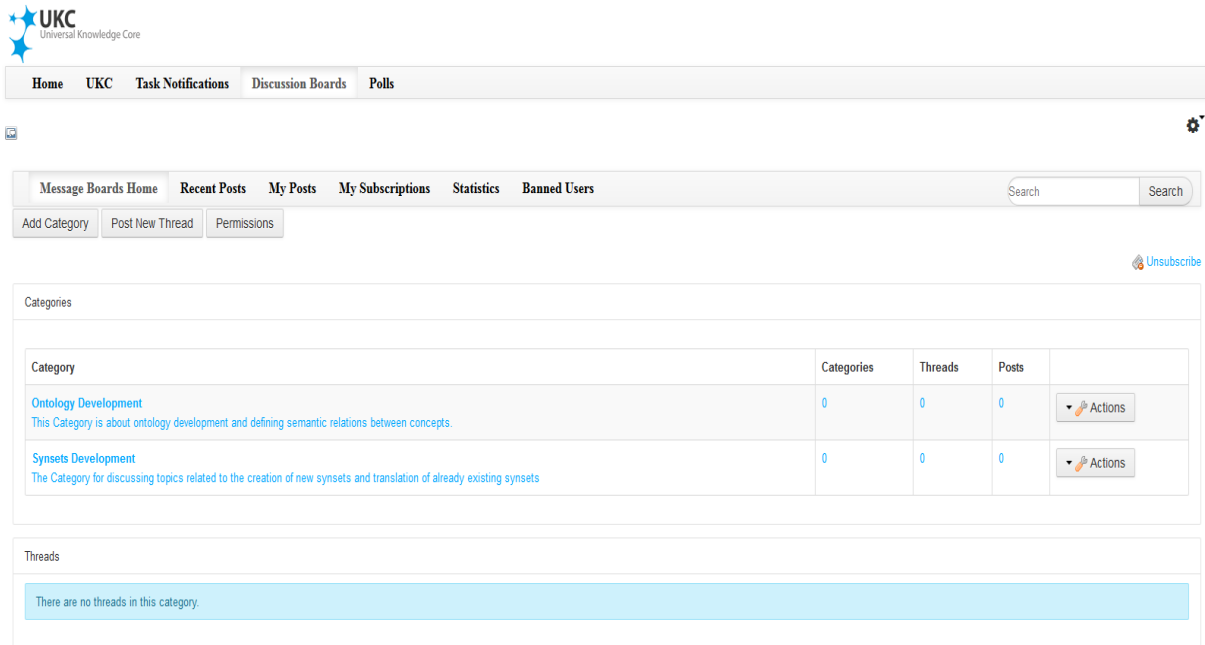
The UKC validator may also share task specific comments and opinions with the UKC Developer and other UKC Validators.

The screenshot displays the 'Review: Synset Entry' task validation interface. At the top, the task title 'Review: Synset Entry' is shown with a plus icon. Below this, the 'Assigned to' field is set to 'Ahmed Tawfik' with an '- Assign to...' link. The 'State' is 'Review', accompanied by a document icon with two red checkmarks and the label 'Review'. The 'Create Date' is '9/30/14 7:16 PM'. The 'Due Date' is 'Never' with a '- (Change)' link. To the right of these fields are four action buttons: 'Approve', 'Reject', 'Assign to...', and 'Update Due Date'. Below the task details is a preview window for the 'Synset Entry' with the following content:

```
Preview of model.resource.it.unitn.disi.web.core.kb.model.vocabularies.Synset
Synset Entry
Operation: Update
Synset Details:
Language: English
(adjective ) [ kind, tolerant ]
```

Figure 8.5 Task Validation Page

Figure 8.6 shows the discussion boards tab selected. Discussion threads are displayed by category or separately. Users are allowed to take several actions in the board including adding new category of discussions, creating new threads or deleting an existing thread.



UKC
Universal Knowledge Core

Home UKC Task Notifications Discussion Boards Polls

Message Boards Home Recent Posts My Posts My Subscriptions Statistics Banned Users Search Search

Add Category Post New Thread Permissions

Unsubscribe

Categories

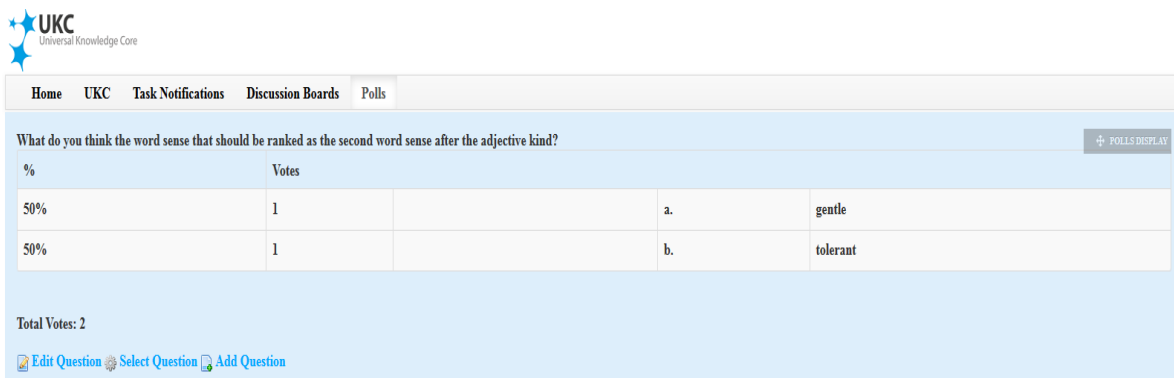
Category	Categories	Threads	Posts	
Ontology Development <small>This Category is about ontology development and defining semantic relations between concepts.</small>	0	0	0	Actions
Synsets Development <small>The Category for discussing topics related to the creation of new synsets and translation of already existing synsets</small>	0	0	0	Actions

Threads

There are no threads in this category.

Figure 8.6 Discussion Boards

Figure 8.7 shows the discussion polls tab selected. In the figure, only one linguistic poll has been created which got a total number of two votes. Users who didn't submit a vote are allowed to vote and view the result. Knowing that, only UKC Administrators are allowed to create new interactive pools



UKC
Universal Knowledge Core

Home UKC Task Notifications Discussion Boards Polls

What do you think the word sense that should be ranked as the second word sense after the adjective kind? POLLS DISPLAY

%	Votes		
50%	1	a.	gentle
50%	1	b.	tolerant

Total Votes: 2

Edit Question Select Question Add Question

Figure 8.7 Interactive Polls

Chapter 9

COLLABORATIVE DEVELOPMENT EXPERIMENT

In this chapter, we present a collaborative multilingual ontology design experiment. The experiment aims to develop a “flowing bodies of water” ontology which is composed of around 70 concepts. The ontology is designed in English by defining the corresponding synsets in English together with the language-independent concepts and semantic relations between them. Then Translating the English synsets to Italian and providing the “flowing bodies of water” ontology localized in Italian.

The chapter starts by explaining the approach that we followed in developing the ontology, we used DERA framework (Giunchiglia et al. 2013), a faceted knowledge representation approach for the development of domain ontologies. The chapter then proceeds by presenting an early experiment which was done following DERA framework using EasyChair¹⁰ as a collaborative tool. In this part of the chapter, we point out to the main outcomes of this early experiment and the shortcoming that we have identified during the collaborative development process. Finally we explain the final evaluation experiment which was done using our collaborative development framework and the results of the experiment.

ONTOLOGY DEVELOPMENT METHODOLOGY: DERA FRAMEWORK

DERA (Domain, Entities, Relations, and Attributes) is a faceted knowledge representation methodology for the development of domain ontologies. DERA provides an approach for describing and reasoning about entities (E) of a specific domain (D). It also accounts for defining the entity attributes (A) and the relations (R) between entities.

¹⁰ www.easychair.org

DERA defines a **domain** as a comprehensive knowledge about a specific topic of interest such as a field of study (e.g. artificial intelligence, computer networks), a discipline (e.g. computer science, or telecommunications), or any real life topic of interest (e.g. space, astronomy, or music).

DERA provides means for organizing the domain into facets, each facet covers a different aspect of the domain. Therefore, we may define a **facet** to be a hierarchy of homogeneous terms describing an aspect of the domain, where each term in the hierarchy denotes a primitive atomic language-independent concept. Facets could be further divided into detailed subfacets.

By following the definitions of domain and facet proposed by DERA, we can define DERA domain ontology as triple: $D = \langle E, R, A \rangle$, Where:

- **E (Entity):** Entity facets are built of homogenous classes of entities, whose instances have either perceptual or conceptual existence. Terms in these hierarchies are explicitly connected by *is-a* or *part-of* relation.
- **R (Relation):** Relation facets are built of homogenous terms representing the relations between entities. Terms in these hierarchies are explicitly connected by *is-a* relation.
- **A (Attribute):** Attribute facets are built of homogenous terms denoting qualitative/quantitative or descriptive properties of entities. We distinguish between attribute names (i.e. length) and attribute values (i.e. long or short) such that each attribute name is associated with a set of corresponding attribute values. Attribute names are connected by *is-a* relation, while attribute values are connected to corresponding attribute names by *value-of* relation.

We argue that DERA framework is characterized by a unique set of features:

1. **Extensibility:** domains, facets, and terms can be added anytime.

2. **Independency:** any domain can be used independently for navigation purposes or by an external application.
3. **Reasoning:** DERA allows for direct mapping to description logic (DL), therefore inherits all the properties and features of DL such as soundness, decidability and decision procedures. The mapping of E/R/A to DL should be obvious. is-a, part-of and value-of relation from the backbone of facets, are assumed to be transitive and asymmetric, and therefore hierarchical. Other relations not having such properties are said to be associative and connect terms in different facets. The result of connected facets constitutes the Tbox of a descriptive ontology.

Ontology development according to DERA methodology should follow a minimal set of guiding principles, described in-detail in (Giunchiglia et al. 2012a). Here we explain them briefly:

1. **Identification of the atomic concepts:** terms in natural (e.g in English) are collected, examined and disambiguated into atomic language-independent concepts. The knowledge developer may collect the terms by reading books, checking online resources, interviewing domain experts, or using any other knowledge acquisition method. Terms with the same meaning (synonyms) are grouped together forming a synset and are given a natural language description (gloss) that makes explicit the intended meaning of the atomic concept. For instance, we can recognize that in English the terms “brook” and “creek” are synonyms forming an atomic concept whose meaning can be described as “a water stream that is smaller than a river”
2. **Analysis:** The atomic concepts are analyzed in order to identify their commonalities and their differences. The main goal of this step is to identify the characteristics of the defined atomic concepts. For instance, we can recognize that for the concept “river we can identify the following characteristics: (1) a flowing body of water, (2) has no fixed boundary, (3) confined within a bed and stream banks, and (4) larger than a brook.

3. **Synthesis:** Concepts are arranged into facets hierarchies such that at each level of the hierarchy the concept represents a different level of abstraction. Child concepts are connected to their parent concept through an explicit is-a (genus-species) or part-of (whole-part) relation. For instance, we can recognize that under the “flowing body of water” facet, the concept of “stream” is-a “flowing body of water” and that is due to their common characteristics. Then we can declare that a “river” is-a “stream” and a “brook” is-a “stream”. Also we can declare that a “stream bend” is part-of a “stream” and “section of stream” is part-of stream.
4. **Standardization:** Each language-independent atomic concept can be potentially denoted with any of the collected terms in the group of synonyms. When the group contains more than one term, a standard (or preferred) term should be selected among the synonyms. It’s recommended to choose the term which is most commonly used in the domain and which minimizes the ambiguity. For instance, the term “brook” is preferred to “creek” to denote the concept of “a water stream that is smaller than a river”
5. **Ordering:** Concepts in each level of the facet hierarchy array are ordered. There are several criteria that could be used in ordering the atomic concepts. They include by chronological order, by spatial order, by increasing and decreasing quantity, by increasing complexity, or any other ordering method related to the domain.

EARLY EXPERIMENT

We performed an early experiment at the purpose of collecting useful requirements for our collaborative platform. The experiment focused on ontology development in English. In the following subsections, we present the experimental setting and results of the experiment.

EXPERIMENTAL SETTING

We aimed at the development of an ontology of “*flowing bodies of water*” including concepts like river and fiord. Candidates were taken from the GeoWordNet ontology (Giunchiglia et al. 2010b) that is an ontology generated by the integration of WordNet with GeoNames¹¹. We followed a peer-review approach carried out by one developer and three different reviewers who had to decide about the acceptance or rejection of the submitted candidate terms in a way similar to the paper review process for conferences.

We used EasyChair to moderate the assignment and review phases. More in detail, EasyChair was used to support the analysis phase of the DERA development where each synset and corresponding gloss was provided by one developer in one file as a submitted paper, accompanied by a detailed explanation of the rationale behind such synset (word senses, gloss, and POS) and commented by the reviewers who could either accept or reject it. In both cases the reviewers provided feedback and typically suggested modifications to the gloss and/or complained on the rationale.

RESULTS OF THE EXPERIMENT

- **Participants Satisfaction:** Developers and Validators felt comfortable to how the assignments and notification of tasks were taking place using EasyChair. In addition, the exchange of comments among participants was done in a seamless way. However, developers spent a long time in the development process as they had to perform the analysis phase using external resources and tools (such as WordNet, and GeoWordNet) then define the newly created synsets and concepts in a new file. The development process required switching between the external resources and the new file several times during the development of each synset which consumed a significant amount of time and effort.
- **Advantages and limitations of EasyChair:** Concerning the advantages, we found out that EasyChair nicely supports the assignment, collection and moderation of

¹¹ <http://www.geonames.org/>

the reviews; it partially supports communication between participants via email facilities; it helps converging to commonly agreed decisions. However, EasyChair is not properly designed for ontology development and validation, but rather for paper review. We identified the following weaknesses:

1. **Pull vs. Push approach:** it is based on a pull (authors submit) rather than push (developers are assigned a task) approach.
2. **Static Workflow:** the workflow is static and cannot be changed. It does not support continuous refinement loops, but only up to one rebuttal phase. In case of rejection from the reviewers, a synset can be resubmitted, but it is hard to keep track of how the submission and the reviews evolve (i.e., what has been changed by the developer with the refinement? Did the developer accommodate for the feedback received?).
3. **Levels of development/validation:** it does not provide a broad view of the implications of an acceptance, i.e. the position that a certain concept would take in the ontology if accepted (w.r.t. the parent, the siblings and the children). In fact, EasyChair can be used to only support the DERA analysis, and not the synthesis (i.e., we cannot get an overview of how the facet is overall getting shape).
4. **Order of development/validation:** given that deeper nodes are defined in terms of higher nodes, the order of review should be top-down, i.e. from the root to the leaves (and not in the order of submission); the tool does not give any suggestion about the order.
5. **Cost of the process:** the process is too costly in terms of time. Everything was submitted and resubmitted as document attachments. This turned out to be impractical as it took significant time and it is not even possible to reconstruct the sequence of submissions as new ones override old ones.
6. **Reputation:** there is an issue of appropriately engaging developers and validators. EasyChair does not support the possibility to maintain a social network of experts to be allocated on demand on the basis of their skills.

FINAL EXPERIMENT

We performed a final experiment at the purpose of evaluating the collaborative platform. The experiment focused on ontology development in English and ontology localization in Italian. In the following subsections, we present the experimental setting and results of the experiment.

EXPERIMENTAL SETTING

We aimed at the development of a multilingual ontology of “*flowing bodies of water*” including concepts like river and fiord. The development process was done in English using the UKC collaborative platform explained in chapter 8. The single approval workflow was used to develop the synsets (one developer and one validator). The approved English synset was assigned to an Italian translator who was obtaining the corresponding Italian synsets from Italian part of MultiWordNet which is then got validated by an Italian native speaker. Figure 9.1 shows the collaborative workflow used to manage the whole process.

In this experiment, discussion threads were initiated between participants and external knowledge experts who were not taking part in the development/validation activities but they were allowed to participate in the discussions and share their opinions. In case of conflicts, an interactive poll was created by an administrator, results of the poll were useful to the validator in taking the final decision.

The following are the list of participants and their roles:

- ***Ontology Developer:*** Defines new synsets and define semantic relations between their language-independent atomic concepts according to DERA methodology guiding principles.
- ***Ontology Validator:*** Validates the defined synsets and the overall ontology according to DERA methodology guiding principles in a collaborative manner with the ontology developer.

- **Italian Translator:** Obtains the corresponding synset in Italian from the Italian part of MultiWordNet, or translate the English synset incase the Italian synset is undefined.
- **Italian Linguistic Validator:** Validates the translated synsets by the Italian translator.
- **Administrator:** Defines the collaborative workflow, user roles, and creates interactive polls in case of conflicts.
- **Community Members:** Their participation is optional, they are able to watch/participate in the on-going discussions and the interactive polls in case of conflicts

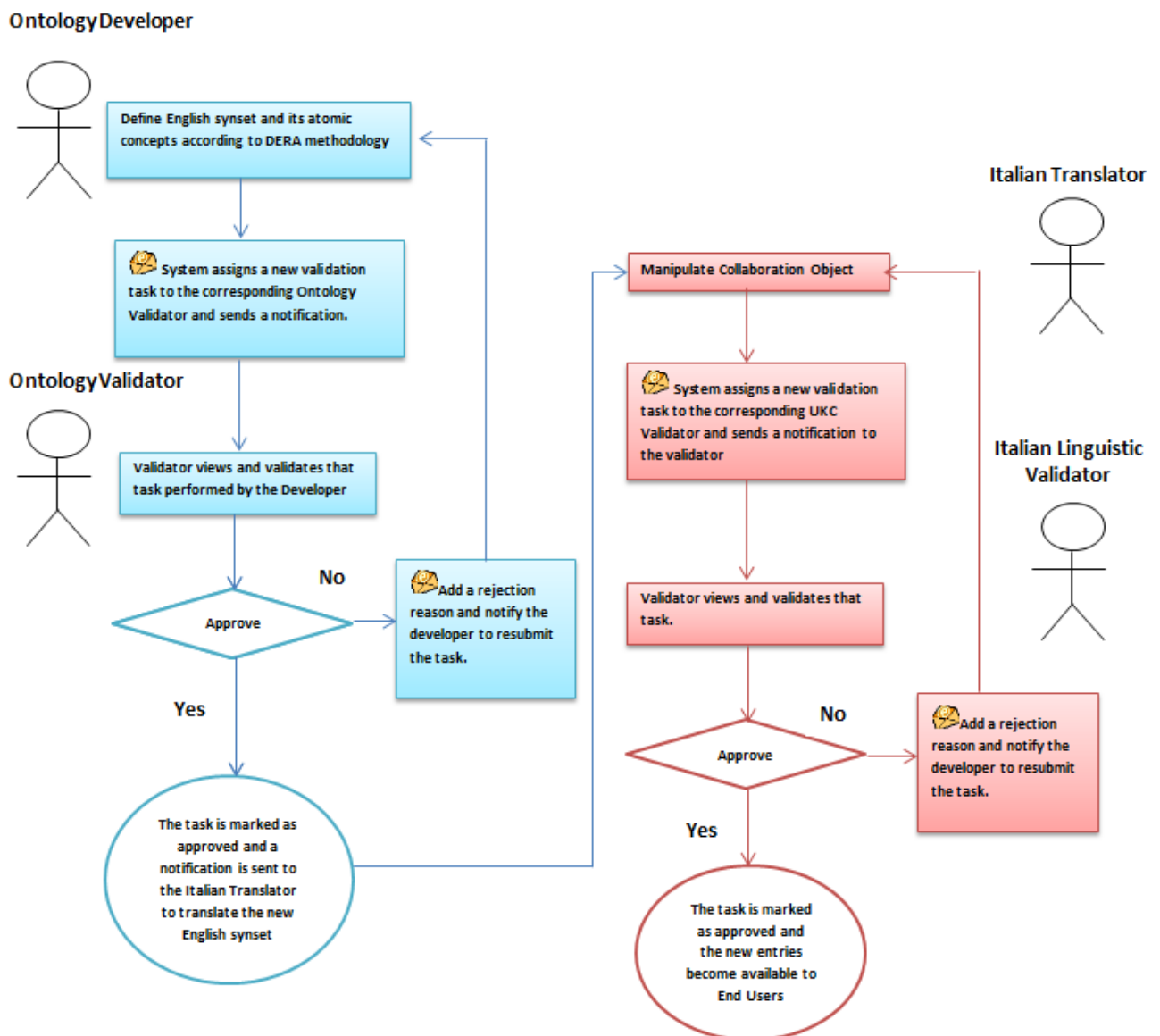


Figure 9.1- Collaborative workflow used in the development of “flowing body of water” multilingual ontology.

RESULTS OF THE EXPERIMENT

- **Participants Satisfaction:** Developers and Validators found the interactive features of the application very pleasing and easy to use. Among these features, interactive input forms, drag and drop facilities, and filtering concepts by semantic relations. These features were used extensively by the participants and saved a significant portion of their time. However, participants requested a new feature related to the sysnet update operation, they preferred to perform multiple synset updates while keeping track of all the previous values.
- **Multilingual Ontology:** the participants produced an ontology composed of 73 concepts (43 entity classes, 19 relations and 11 attributes). As the ontology was developed according to the DERA methodology, the domain ontology (D) is composed of three facets; a facet defining classes of entities (E), a facet defining possible relations between entities (R), and a facet defining entity attributes (A).

Figure 9.2, shows a screen shot obtained from the collaborative platform for the developed “flowing body of water” facet in English and Italian. The facet hierarchy represents the defined classes of entities that could be considered as flowing bodies of water. At the top of the hierarchy, the concept of “flowing body of water” which has one is-a child concept “stream”. The “stream” concept has many child concepts; few concepts are connected by part-of relation such as “stream-bend” and “section of stream” while the rest of the concepts are connected by is-a relation such as “river”, “brook” and “spring”.

The Italian translator has identified two lexical gaps in the Italian language billabong and oxbow. The former refers to a branch of flowing water made by the portion of water coming from the main stream only when the water level is high. The later refers to a U-shaped curve in a stream.

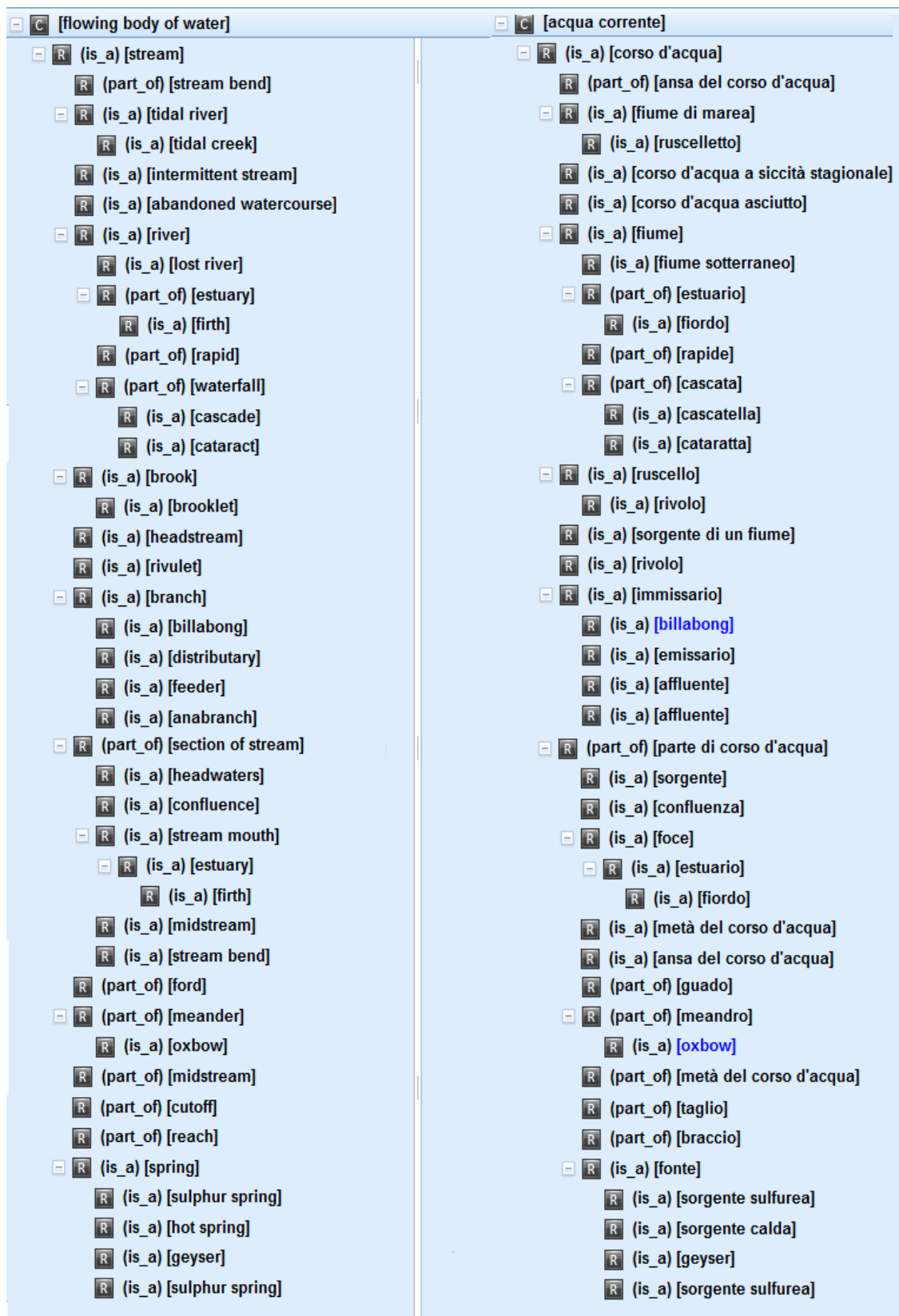


Figure 9.2- Flowing Bodies of Water Ontology (Classes of Entities) in English (left) and Italian (right)

Figure 9.3, shows a screen shot of the possible relations between entities in English and Italian. The concepts: “branch”, “direction” and “relative level” are among the possible relations that may exist between flowing bodies of water.

<p>[branch]</p> <ul style="list-style-type: none"> R (is_a) [billabong] R (is_a) [distributary] R (is_a) [feeder] R (is_a) [anabranh] 	<p>[immissario]</p> <ul style="list-style-type: none"> R (is_a) [billabong] R (is_a) [emissario] R (is_a) [affluente] R (is_a) [affluente]
<p>[direction]</p> <ul style="list-style-type: none"> R (is_a) [east] R (is_a) [south] R (is_a) [west] R (is_a) [north] R (is_a) [southeastern] R (is_a) [southwestern] R (is_a) [northwestern] R (is_a) [northeastern] R (is_a) [incoming] R (is_a) [outgoing] 	<p>[direzione]</p> <ul style="list-style-type: none"> R (is_a) [est] R (is_a) [sud] R (is_a) [ovest] R (is_a) [nord] R (is_a) [sudorientale] R (is_a) [sudoccidentale] R (is_a) [nordoccidentale] R (is_a) [nordorientale] R (is_a) [entrante] R (is_a) [uscente]
<p>[relative level]</p> <ul style="list-style-type: none"> R (is_a) [below] R (is_a) [above] 	<p>[locuzione di luogo]</p> <ul style="list-style-type: none"> R (is_a) [sotto] R (is_a) [sopra]

Figure 9.3- Flowing Bodies of Water Ontology (Relations between Entities) in English (left) and Italian (right)

Figure 9.4, shows a screen shot for the entity attributes in English and Italian. The concepts: “name”, “depth”, “length”, “area”, “latitude”, “longitude” and “altitude” are the attributes defined for any flowing body of water. The depth attribute name is related to deep and shallow attribute values via value-of relation and the length attribute name is related to long and short attribute values via value-of relation.

C [name]	C [nome]
C [depth]	C <28288> [profondità]
R (value_of) [deep]	R <85644> (value_of) [profondo]
R (value_of) [shallow]	R <85647> (value_of) [poco profondo]
C <28259> [length]	C <28259> [lunghezza]
R <89775> (value_of) [long]	R <89775> (value_of) [lungo]
R <89763> (value_of) [short]	R <89763> (value_of) [corto]
C [area]	C [area di superficie]
C [latitude]	C [latitudine]
C [longitude]	C [longitudine]
C [altitude]	C [altitudine]

Figure 9.4- Flowing Bodies of Water Ontology (Entity Attributes) in English (left) and Italian (right)

Chapter 10

CONCLUSION AND FUTURE WORK

The UKC collaborative platform is an effective linguistic resource enrichment and ontology development tool that allows for knowledge organization in a language-independent manner. The collaborative development process takes place in a customizable community based manner following a workflow-based approach and making use of web 2.0 social and collaborative features.

The contributions presented in this thesis have been considered as a basic practical step in which the guidelines for an ideal and usable system have been set, along with its architecture, but the features yet to be implemented are considered as a part of the future work. Among these, introducing more customizable collaborative workflows and introducing two main types of collaborative objects: entity types and domains.

COLLABORATIVE DEVELOPMENT OF ENTITY TYPES

Entity Types development aims to extend the platform and allows for collaborative development of the codifying information about *entities*. Basically, it's essential to understand that an *entity* is any self-contained real world object denoted by its identifier or name and has its own set of attributes. For instance, Paris is an entity of type *Location*. An *entity type* provides a sort of template for defining entity attributes and relations between entity types using language-independent concepts.

In order to clarify the idea in more detail, we can make an analogy between the entity types and the Object Oriented Programming methodology, an entity type definition analogous a class and an entity analogous an instance object or an instance of class. Other object oriented features like inheritance are also supported, entity type may extend a more generic entity type. Figure-10.1 gives an example of four entity type definitions (Person, Politician, Organization, and Political Party) modeled as a class

diagram. Politician inherits Person as it is considered as a more specific entity type definition of Person and Political Party inherits Organization as it's considered as a more entity type of Organization. Entities of type Politician are related to the Political Party entities, the relation type could be defined as “belongs to”.

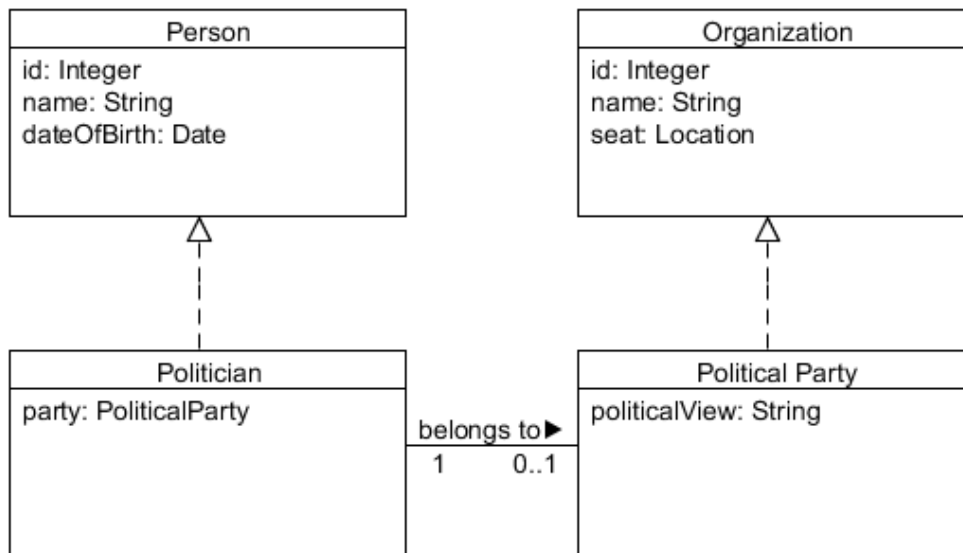


Figure-10.1: Entity type definitions modeled as UML class diagram

COLLABORATIVE DEVELOPMENT OF DOMAINS

Domain development aims to extend the platform and allows for collaborative development of the codifying information about *domains*. Basically, a *domain* provides a comprehensive knowledge about a specific topic of interest such as a field of study (i.e. Computer Science, Physics etc.) or any real life topic of interest (i.e. Space, Sports, movies etc.).

A domain could be defined as an aggregation of entity types. The purpose of a domain is to define the basic terminology of a specific domain of interest in terms of relevant entity classes. The domain terminology further defines and represents the relation between entity types, and entity type attributes using language independent concepts.

Figure-10.2 gives an example for the list of entity types associated with the *movie domain* together with its corresponding domain terminology. Domains could be aggregated to compose a *hierarchy of domains* where the low level domains are more specific than the upper level domain. For instance, the domain soccer can be defined as more specific domain than Sports.

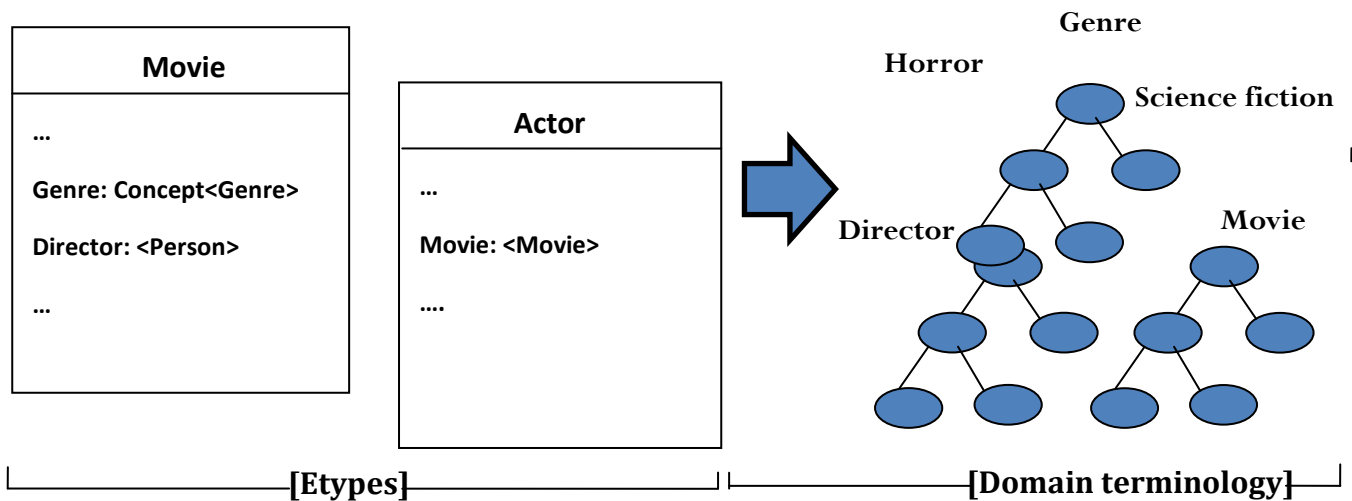


Figure-10.2: Movie Domain defined as an aggregation of entity types (Movie and Actor) and the corresponding domain terminology as language-independent concepts.

Bibliography

- [1] Auer, S., Dietzold, S., & Riechert, T. (2006). OntoWiki—a tool for social, semantic collaboration. In *The Semantic Web-ISWC 2006* (pp. 736-749). Springer Berlin Heidelberg.
- [2] Baader, F. (Ed.). (2003). *The description logic handbook: theory, implementation, and applications*. Cambridge university press.
- [3] Babic, F., Wagner, J., & Paralic, J. (2008, January). The role of ontologies in collaborative systems. In *Applied Machine Intelligence and Informatics, 2008. SAMI 2008. 6th International Symposium on* (pp. 119-124). IEEE.
- [4] Baker, C. F., Fillmore, C. J., & Cronin, B. (2003). The structure of the FrameNet database. *International Journal of Lexicography*, 16(3), 281-296.
- [5] Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998, August). The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1* (pp. 86-90). Association for Computational Linguistics.
- [6] Bannon, L. J., & Bødker, S. (1989). Beyond the interface: Encountering artifacts in use. *DAIMI Report Series*, 18(288).
- [7] Bao, J., & Honavar, V. (2004). Collaborative ontology building with wiki@ nt. In *3rd Intl. Workshop on Evaluation of Ontology Based Tools at Intl. Semantic Web Conference*.
- [8] Bardram, J. (2000). Scenario-based design of cooperative systems. *Group decision and negotiation*, 9(3), 237-250.
- [9] Boas, H. C. (2002). *Bilingual FrameNet Dictionaries for Machine Translation*. In LREC.
- [10] Bodker, S. (2000). Scenarios in User-Centered Design - Setting the Stage for Reflection and Action. *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences*.

- [11] Bødker, S., Nielsen, C., & Petersen, M. G. (2000, August). Creativity, cooperation and interactive design. In Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques (pp. 252-261). ACM.
- [12] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008, June). Freebase: a collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data (pp. 1247-1250). ACM.
- [13] Bond, F., & KYONGHEE, P. (2012). A Survey of WordNets and their Licenses. In Proceedings of the 6th International Global WordNet Conference (pp. 64-71).
- [14] Bond, F., & Foster, R. (2013). Linking and Extending an Open Multilingual Wordnet. In ACL (1) (pp. 1352-1362).
- [15] Borst, W. N. (1997). Construction of engineering ontologies for knowledge sharing and reuse. Universiteit Twente
- [16] Bouquet, P., & Giunchiglia, F. (1995). Reasoning about theory adequacy. a new solution to the qualification problem. *Fundamenta Informaticae*, 23(2), 247-262.
- [17] Braun, S., Schmidt, A. P., Walter, A., Nagypal, G., & Zacharias, V. (2007). Ontology Maturing: a Collaborative Web 2.0 Approach to Ontology Engineering. *Ckc*, 273.
- [18] Buitelaar, P., Cimiano, P., Haase, P., & Sintek, M. (2009). Towards linguistically grounded ontologies. In *The semantic web: research and applications* (pp. 111-125). Springer Berlin Heidelberg.
- [19] Buitelaar, P., Sintek, M., & Kiesel, M. (2006). *A multilingual/multimedia lexicon model for ontologies* (pp. 502-513). Springer Berlin Heidelberg.
- [20] Burchardt, A., Erk, K., & Frank, A. (2005). A WordNet detour to FrameNet. *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen*, 8, 408-421.
- [21] COMPOSER, T. (2007). TOPBRAID COMPOSER 2007 Features and getting Started Guide Version 1.0, created by TopQuadrant, US
- [22] Carroll, J. M., Rosson, M. B., Convertino, G., & Ganoë, C. H. (2006). Awareness and teamwork in computer-supported collaborations. *Interacting with computers*, 18(1), 21-46.

- [23] Corcho, O., Fernández-López, M., & Gómez-Pérez, A. (2003). Methodologies, tools and languages for building ontologies. Where is their meeting point?. *Data & knowledge engineering*, 46(1), 41-64.
- [24] De Moor, A. (2005, June). Ontology-guided meaning negotiation in communities of practice. In *Proc. of the Workshop on the Design for Large-Scale Digital Communities at the 2nd International Conference on Communities and Technologies (C&T 2005)*, Milano, Italy.
- [25] Derwojedowa, M., Piasecki, M., Szpakowicz, S., Zawisławska, M., & Broda, B. (2008, January). Words, concepts and relations in the construction of Polish WordNet. In *Proceedings of the Global WordNet Conference*, Seged, Hungary (pp. 162-177).
- [26] Dewan, P. (1999). Architectures for collaborative applications. *Computer Supported Co-operative Work*, 7, 169-193.
- [27] Dewan, P. (2001). An integrated approach to designing and evaluating collaborative applications and infrastructures. *Computer Supported Cooperative Work (CSCW)*, 10(1), 75-111.
- [28] Ding, L., Kolari, P., Ding, Z., & Avancha, S. (2007). Using ontologies in the semantic web: A survey. In *Ontologies* (pp. 79-113). Springer US.
- [29] Dutta, B., Giunchiglia, F. and Maltese, V. (2012). A facet-based methodology for geo-spatial modelling. *Journal on Data Semantics* May 2012, Volume 1, Issue 1, pp 57-73.
- [30] Eckle-Kohler, J., Gurevych, I., Hartmann, S., Matuschek, M., & Meyer, C. M. (2012, May). UBY-LMF-A Uniform Model for Standardizing Heterogeneous Lexical-Semantic Resources in ISO-LMF. In *LREC* (pp. 275-282).
- [31] Espinoza, M., Gómez-Pérez, A., & Montiel-Ponsoda, E. (2009). Multilingual and localization support for ontologies. In *The Semantic Web: Research and Applications* (pp. 821-825). Springer Berlin Heidelberg.
- [32] Espinoza, M., Montiel-Ponsoda, E., & Gómez-Pérez, A. (2009, September). Ontology localization. In *Proceedings of the fifth international conference on Knowledge capture* (pp. 33-40). ACM.
- [33] Falconer, S., Tudorache, T., & Noy, N. F. (2011, June). An analysis of collaborative patterns in large-scale ontology development projects. In *Proceedings of the sixth international conference on Knowledge capture* (pp. 25-32). ACM.

- [34] Farquhar, A., Fikes, R., & Rice, J. (1997). The ontolingua server: A tool for collaborative ontology construction. *International journal of human-computer studies*, 46(6), 707-727.
- [35] Fellbaum, C. (1998). *WordNet*. Blackwell Publishing Ltd.
- [36] Fensel, D. (2001). *Ontologies: A Silver Bullet for Knowledge Management and Electronic Management*.
- [37] Fensel, D., Leiter, B., Thaler, S., Thalhammer, A., & Toma, I. (2012, September). Effective and Efficient On-Line Communication. In *DEXA Workshops* (pp. 294-298).
- [38] Fillmore, C. J., Baker, C. F., & Sato, H. (2002). The FrameNet Database and Software Tools. In *LREC*.
- [39] Fischer, G. (2009). End-user development and meta-design: Foundations for cultures of participation. In *End-user development* (pp. 3-14). Springer Berlin Heidelberg.
- [40] Fischer, G. (2011). Understanding, fostering, and supporting cultures of participation. *interactions*, 18(3), 42-53.
- [41] Fluit, C., Sabou, M., & Van Harmelen, F. (2006). Ontology-based information visualization: toward semantic web applications. In *Visualizing the semantic web* (pp. 45-58). Springer London.
- [42] Fontana, A., & Frey, J. H. (2000). The interview: From structured questions to negotiated text. *Handbook of qualitative research*, 2, 645-672.
- [43] Gasson, S. (2003). Human-centered vs. user-centered approaches. Drexel University. College of Information Science and Technology. Faculty Publications and Research.
- [44] Ganbold, A., Farazi, F., & Giunchiglia, F. (2014, March). An Experiment in Managing Language Diversity Across Cultures. In *eKNOW 2014, The Sixth International Conference on Information, Process, and Knowledge Management*(pp. 51-57).
- [45] Gangemi, A. (2005). Ontology design patterns for semantic web content. In *The Semantic Web–ISWC 2005* (pp. 262-276). Springer Berlin Heidelberg.
- [46] García-Peñalvo, F. J., Colomo-Palacios, R., García, J., & Therón, R. (2012). Towards an ontology modeling tool. A validation in software engineering scenarios. *Expert Systems with Applications*, 39(13), 11468-11478.

- [47] Ghidini, C., & Giunchiglia, F. (2003). A semantics for abstraction.
- [48] Ghidini, C., Rospocher, M., & Serafini, L. (2010). Moki: a wiki-based conceptual modeling tool. *ISWC 2010 Posters & Demonstrations Track: Collected Abstracts*, 658, 77-80.
- [49] Ghidini, C., Rospocher, M., & Serafini, L. (2012, January). Conceptual modeling in wikis: a reference architecture and a tool. In *eKNOW 2012, The Fourth International Conference on Information, Process, and Knowledge Management* (pp. 128-135).
- [50] Giunchiglia, F. and Dutta, B. (2011). DERA: A Faceted Knowledge Organization Framework. *International Conference on Theory and Practice of Digital Libraries*.
- [51] Giunchiglia, F., Dutta, B., & Maltese, V. (2009a). Faceted lightweight ontologies. In *Conceptual Modeling: Foundations and Applications* (pp. 36-51). Springer Berlin Heidelberg.
- [52] Giunchiglia, F., Dutta, B., & Maltese, V. (2013). From Knowledge Organization to Knowledge Representation.
- [53] Giunchiglia, F., Dutta, B. and Maltese, V. and Farazi, F. (2012a). A facet-based methodology for the construction of large-scale geospatial ontology. *Journal on Data Semantics (JoDS)*, Vol. 1, Issue 1 (2012), pp. 57-73.
- [54] Giunchiglia, F., Maltese, V., & Autayeu, A. (2012). Computing minimal mappings between lightweight ontologies. *International Journal on Digital Libraries*, 12(4), 179-193.
- [55] Giunchiglia, F., Maltese, V., B. Dutta (2012b). Domains and Context: First steps towards managing diversity in knowledge. *Journal of Web Semantics (JWS)* Vol 12 - 13 (2012): Special Issue: Reasoning with Context in the Semantic Web.
- [56] Giunchiglia, F., Maltese, V., Farazi, F., & Dutta, B. (2010b). GeoWordNet: a resource for geo-spatial applications. In *The Semantic Web: Research and Applications* (pp. 121-136). Springer Berlin Heidelberg.
- [57] Giunchiglia, F., Marchese, M., & Zaihrayeu, I. (2005, May). Towards a theory of formal classification. In *Proceedings of the AAAI-05 Workshop on Contexts and Ontologies: Theory, Practice and Applications (C&O-2005)*, Pittsburgh, Pennsylvania, USA (pp. 1-8).

- [58] Giunchiglia, F., Marchese, M., & Zaihrayeu, I. (2007). Encoding classifications into lightweight ontologies. In *Journal on data semantics VIII* (pp. 57-81). Springer Berlin Heidelberg.
- [59] Giunchiglia, F., McNeill, F., Yatskevich, M., Pane, J., Besana, P., & Shvaiko, P. (2008). Approximate structure-preserving semantic matching. In *On the Move to Meaningful Internet Systems: OTM 2008* (pp. 1217-1234). Springer Berlin Heidelberg.
- [60] Giunchiglia, F., Villafiorita, A., & Walsh, T. (1997). Theories of abstraction. *AI communications*, 10(3), 167-176.
- [61] Giunchiglia, F., & Walsh, T. (1992). A theory of abstraction. *Artificial Intelligence*, 57(2), 323-389.
- [62] Giunchiglia, F., Yatskevich, M., & McNeill, F. (2007). Structure preserving semantic matching.
- [63] Giunchiglia, F., Yatskevich, M., & Shvaiko, P. (2007a). Semantic matching: Algorithms and implementation. In *Journal on Data Semantics IX* (pp. 1-38). Springer Berlin Heidelberg.
- [64] Giunchiglia, F., & Zaihrayeu, I. (2009b). Lightweight ontologies. In *Encyclopedia of Database Systems* (pp. 1613-1619). Springer US.
- [65] Giunchiglia, F., Zaihrayeu, I., & Kharkevich, U. (2007b). Formalizing the get-specific document classification algorithm (pp. 26-37). Springer Berlin Heidelberg.
- [66] Gonzalez-Agirre, A., Laparra E., and Rigau G. (2012). Multilingual central repository version 3.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, pages 2525–2529.
- [67] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2), 199-220. Grudin, J., & Poltrock, S. (2012). Taxonomy and theory in computer supported cooperative work. *The Oxford Handbook of Industrial and Organizational Psychology*. Oxford University Press, New York.
- [68] Gutiérrez, Y and Orquín, A. (2011). Enriching the Integration of Semantic Resources based on WordNet. *Procesamiento del Lenguaje Natural*, vol. 47, pp. 249-257.

- [69] Guarino, N., Oberle, D., & Staab, S. (2009). What is an Ontology?. In Handbook on ontologies (pp. 1-17). Springer Berlin Heidelberg.
- [70] Hanoka, V., & Sagot, B. (2012). Wordnet creation and extension made simple: A multilingual lexicon-based approach using wiki resources. In Proceedings of the 8th international conference on Language Resources and Evaluation (LREC 2012).
- [71] Hinds, P., & McGrath, C. (2006, November). Structures that work: social structure, work structure and coordination ease in geographically distributed teams. In Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work (pp. 343-352). ACM.
- [72] Hoffart, J., Suchanek, F. M., Berberich, K., Lewis-Kelham, E., De Melo, G., & Weikum, G. (2011, March). YAGO2: exploring and querying world knowledge in time, space, context, and many languages. In Proceedings of the 20th international conference companion on World wide web (pp. 229-232). ACM.
- [73] Hoffart, J., Suchanek, F. M., Berberich, K., & Weikum, G. (2013). YAGO2: a spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194, 28-61.
- [74] Howard, T. (2009). *Design to thrive: Creating social networks and online communities that last*. Morgan Kaufmann.
- [75] Isahara, H., Bond, F., Uchimoto, K., Utiyama, M., & Kanzaki, K. (2008). Development of the Japanese WordNet.
- [76] Jentzsch, A. (2009). DBpedia—Extracting structured data from Wikipedia. Presentation at Semantic Web In Bibliotheken (SWIB 2009), Cologne, Germany.
- [77] Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., & Giannopoulou, E. (2007). Ontology visualization methods—a survey. *ACM Computing Surveys (CSUR)*, 39(4), 10.
- [78] Katifori, A., Torou, E., Halatsis, C., Lepouras, G., & Vassilakis, C. (2006, July). A comparative study of four ontology visualization techniques in protege: Experiment setup and preliminary results. In *Information Visualization, 2006. IV 2006. Tenth International Conference on* (pp. 417-423). IEEE.

- [79] Katifori, A., Torou, E., Vassilakis, C., Lepouras, G., & Halatsis, C. (2008, June). Selected results of a comparative study of four ontology visualization methods for information retrieval tasks. In *Research Challenges in Information Science, 2008. RCIS 2008. Second International Conference on* (pp. 133-140). IEEE.
- [80] Khondoker, M. R., & Mueller, P. (2010, March). Comparing ontology development tools based on an online survey. *World Congress on Engineering 2010 (WCE 2010)*, London, UK.
- [81] Koeva, S., Mihov, S., & Tinchev, T. (2004). Bulgarian Wordnet–Structure and Validation. *Romanian Journal of Information Science and Technology*, 7(1-2), 61-78.
- [82] Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Morsey, M., van Kleef, P., Auer, S., & Bizer, C. (2014). DBpedia–A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*.
- [83] Matuszek, C., Cabral, J., Witbrock, M. J., & DeOliveira, J. (2006, March). An Introduction to the Syntax and Content of Cyc. In *AAAI Spring Symposium: Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering* (pp. 44-49).
- [84] Meersman, R. A. (1999). Semantic ontology tools in IS design. In *Foundations of Intelligent Systems* (pp. 30-45). Springer Berlin Heidelberg.
- [85] Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11), 39-41.
- [86] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4), 235-244.
- [87] Milne, D., & Witten, I. H. (2013). An open-source toolkit for mining Wikipedia. *Artificial Intelligence*, 194, 222-239.
- [88] Mizoguchi, R. (2003). Part 1: Introduction to ontological engineering. *New Generation Computing*, 21(4), 365-384.
- [89] Mizoguchi, R. (2004). Part 2: Ontology development, tools and languages. *New Generation Computing*, 22(1), 61-96.
- [90] Mizoguchi, R. (2004). Part 3: Advanced course of ontological engineering. *New Generation Computing*, 22(2), 193-220.

- [91] Montiel-Ponsoda, E., Aguado de Cea, G., Gómez-Pérez, A., & Peters, W. (2008). Modelling multilinguality in ontologies.
- [92] Montiel-Ponsoda, E., Aguado de Cea, G., Gómez-Pérez, A., & Peters, W. (2011). Enriching ontologies with multilingual information. *Natural language engineering*, 17(03), 283-309
- [93] Morsey, M., Lehmann, J., Auer, S., Stadler, C., & Hellmann, S. (2012). Dbpedia and the live extraction of structured data from wikipedia. *Program: electronic library and information systems*, 46(2), 157-181.
- [94] Muñoz-García, Ó., Gómez-Pérez, A., Iglesias-Sucasas, M., & Kim, S. (2007). A Workflow for the Networked Ontologies Lifecycle: A Case Study in FAO of the UN. In *Current Topics in Artificial Intelligence* (pp. 200-209). Springer Berlin Heidelberg.
- [95] Nastase, V., & Strube, M. (2013). Transforming Wikipedia into a large scale multilingual concept network. *Artificial Intelligence*, 194, 62-85.
- [96] Navigli, R. (2012). BabelNet goes to the (Multilingual) Semantic Web. In *MSW*.
- [97] Navigli, R. and Ponzetto, SP. (2010) BabelNet: Building a very large multilingual semantic network. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, 11–16 July 2010, pp. 216–225.
- [98] Navigli, R. and Ponzetto, SP. (2012) BabelNetXplorer: A Platform for Multilingual Lexical Knowledge Base Access and Exploration. *Proceedings of International World Wide Web Conference (IW3C2)*, Lyon, France, 16-20 April 2012.
- [99] Navigli, R., & Ponzetto, S. P. (2012). BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193, 217-250.
- [100] Neale, D. C., Carroll, J. M., & Rosson, M. B. (2004, November). Evaluating computer-supported cooperative work: models and frameworks. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work* (pp. 112-121). ACM.
- [101] Noy, N. F., Chugh, A., & Alani, H. (2008). The CKC challenge: Exploring tools for collaborative knowledge construction. *Intelligent Systems, IEEE*, 23(1), 64-68.
- [102] Noy, N. F., Chugh, A., Liu, W., & Musen, M. A. (2006). A framework for ontology evolution in collaborative environments. In *The Semantic Web-ISWC 2006* (pp. 544-558). Springer Berlin Heidelberg.

- [103] Noy, N. F., & McGuinness, D. L. (2001). *Ontology development 101: A guide to creating your first ontology*.
- [104] Noy, N. F., & Tudorache, T. (2008, March). Collaborative Ontology Development on the (Semantic) Web. In *AAAI Spring Symposium: Symbiotic Relationships between Semantic Web and Knowledge Engineering* (pp. 63-68).
- [105] Noy, N., Tudorache, T., Nyulas, C., & Musen, M. (2010). The ontology life cycle: Integrated tools for editing, publishing, peer review, and evolution of ontologies. In *AMIA Annual Symposium Proceedings* (Vol. 2010, p. 552). American Medical Informatics Association.
- [106] Palma, R., Corcho, O., Gómez-Pérez, A., & Haase, P. (2011). A holistic approach to collaborative ontology development based on change management. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(3), 299-314.
- [107] Palma, R., Haase, P., Corcho, O., Gómez-Pérez, A., & Ji, Q. (2008). An editorial workflow approach for collaborative ontology development. In *The Semantic Web* (pp. 227-241). Springer Berlin Heidelberg.
- [108] Patel, C., Supekar, K., Lee, Y., & Park, E. K. (2003, November). OntoKhoj: a semantic web portal for ontology searching, ranking and classification. In *Proceedings of the 5th ACM international workshop on Web information and data management* (pp. 58-61). ACM.
- [109] Peters, W., Montiel-Ponsoda, E., Aguado de Cea, G., & Gómez-Pérez, A. (2007). Localizing ontologies in OWL.
- [110] Powell, A., Piccoli, G., & Ives, B. (2004). Virtual teams: a review of current literature and directions for future research. *ACM Sigmis Database*, 35(1), 6-36.
- [111] Pruitt, J., & Grudin, J. (2003, June). Personas: practice and theory. In *Proceedings of the 2003 conference on Designing for user experiences* (pp. 1-15). ACM.
- [112] Reinhard, W., Schweitzer, J., Volksen, G., & Weber, M. (1994). CSCW tools: concepts and architectures. *Computer*, 27(5), 28-36.
- [113] Richards, D. (2009). A social software/Web 2.0 approach to collaborative knowledge engineering. *Information Sciences*, 179(15), 2515-2523.
- [114] Rosson, M. B., & Carroll, J. M. (2009). Scenario based design. *Human computer interaction*. Boca Raton, FL, 145-162.

- [115] Ruppenhofer, J., Baker, C. F., & Fillmore, C. J. F. (2003). Collocational information in the FrameNet database. In Proceedings of the Tenth EURALEX International Congress, EURALEX 2002: Copenhagen, Denmark, August 13-17, 2002 (pp. 359-369).
- [116] Salayandia, L., da Silva, P. P., Gates, A. Q., & Salcedo, F. (2006, December). Workflow-driven ontologies: An earth sciences case study. In e-Science and Grid Computing, 2006. e-Science'06. Second IEEE International Conference on (pp. 17-17). IEEE.
- [117] Sebastian, A., Noy, N. F., Tudorache, T., & Musen, M. A. (2008a). A generic ontology for collaborative ontology-development workflows. In Knowledge Engineering: Practice and Patterns (pp. 318-328). Springer Berlin Heidelberg.
- [118] Sebastian, A., Tudorache, T., Noy, N. F., & Musen, M. A. (2008b). Customizable workflow support for collaborative ontology development. In 4th International Workshop on Semantic Web Enabled Software Engineering (SWESE) at ISWC (Vol. 2008).
- [119] Sharp, H., Rogers, Y., & Preece, J. (2011). Interaction design: beyond human-computer interaction.
- [120] Simperl, E. P. B., & Tempich, C. (2006). Ontology engineering: a reality check. In On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE (pp. 836-854). Springer Berlin Heidelberg.
- [121] Simperl, E., & Luczak-Rösch, M. (2014). Collaborative ontology engineering: a survey. *The Knowledge Engineering Review*, 29(01), 101-131.
- [122] Speer, R., & Havasi, C. (2012, May). Representing General Relational Knowledge in ConceptNet 5. In LREC (pp. 3679-3686).
- [123] Spyns, P., Meersman, R., & Jarrar, M. (2002). Data modelling versus ontology engineering. *ACM SIGMod Record*, 31(4), 12-17.
- [124] Staab, S., & Studer, R. (2010). *Handbook on ontologies*. Springer.
- [125] Studer, R., Benjamins, V. R., & Fensel, D. (1998). Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1), 161-197.

- [126] Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., & Wenke, D. (2002). *OntoEdit: Collaborative ontology development for the semantic web* (pp. 221-235). Springer Berlin Heidelberg.
- [127] Tempich, C., Simperl, E., Luczak, M., Studer, R., & Pinto, H. S. (2007). *Argumentation-based ontology engineering*. *IEEE Intelligent Systems*, 22(6), 52-59.
- [128] Tudorache, T., & Noy, N. F. (2007, July). *Collaborative Protege*. In *CKC*.
- [129] Tudorache, T., Noy, N. F., Tu, S., & Musen, M. A. (2008a). *Supporting collaborative ontology development in Protégé* (pp. 17-32). Springer Berlin Heidelberg.
- [130] Tudorache, T., Nyulas, C., Noy, N. F., & Musen, M. A. (2013). *WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the web*. *Semantic web*, 4(1), 89-99.
- [131] Tudorache, T., Vendetti, J., & Noy, N. F. (2008b, October). *Web-Protege: A Lightweight OWL Ontology Editor for the Web*. In *OWLED* (Vol. 432).
- [132] Tufis, D., Mititelu, V. B., Bozianu, L., & Mihaila, C. (2006). *Romanian wordnet: New developments and applications*. In *Proceedings of the 3rd Conference of the Global WordNet Association* (pp. 337-344).
- [133] Valo, A., Hyvönen, E., & Komulainen, V. (2005, September). *A tool for collaborative ontology development for the semantic web*. In *International Conference on Dublin Core and Metadata Applications* (pp. pp-209).
- [134] Vossen, P. (1998). *A multilingual database with lexical semantic networks*. Kluwer Academic Publishers, Dordrecht.
- [135] Wang, X., Chan, C. W., & Hamilton, H. J. (2002, July). *Design of knowledge-based systems with the ontology-domain-system approach*. In *Proceedings of the 14th international conference on Software engineering and knowledge engineering* (pp. 233-236). ACM.
- [136] Wang, Z., Li, J., Wang, Z., & Tang, J. (2012, April). *Cross-lingual knowledge linking across wiki knowledge bases*. In *Proceedings of the 21st international conference on World Wide Web* (pp. 459-468). ACM.
- [137] Weiten, M. (2009). *Ontostudio® as a ontology engineering environment* (pp. 51-60). Springer Berlin Heidelberg.

[138] Zaihrayeu, I., Sun L., Giunchiglia,F., Pan, W., Ju,Q., Chi, M. and Huang X (2007).
From Web Directories to Ontologies: NaturalLanguage Processing Challenges.

APPENDIX A - INTERVIEWS

Interviews can be defined as a conversation with a purpose between an interviewer and interviewee. Interviews could be used as an effective way for gathering requirements before designing and implementing new systems. Fontana and Frey (Fontana, A. et al 2000) listed four main types of interviews: *structured interviews*, *unstructured interviews*, *semi-structured interviews*, and *focus groups*. The first three types differ on the nature of the interview questions and according to how much control the interviewer imposes on the conversation. The fourth type is more like a brainstorming session or an open discussion with a group of people. Here, we define the four main types of interview in more detail:

- 1- ***Structured interviews*** are forms of discussions in which the interviewer asks a set of specific questions similar to those in a questionnaire and requires an answer from a predetermined set of alternatives. For example, did you use any online community for ontology engineering and the expected answer should be either yes or no.
- 2- ***Unstructured interviews*** are more like conversations around a particular topic and involve open questions with no particular expectation about the answer from the interviewee. For example, what are the features that attracted you while using your favorite online community?
- 3- ***Semi-structured interviews*** combine features of structured and unstructured interviews by using both closed and open questions during the interview.
- 4- ***Focus groups*** are forms of open discussions between small groups of people guided by a facilitator.

After reviewing the four different types of interviews, we decided that the best way to interview our knowledge experts is by following an unstructured interviews and let the experts express their knowledge freely while asking them a set of open questions.

The following table has been used during the interviews to gather requirements for UKC WordNet Applications, the table shows sample of the predefined questions. Interviews were also accompanied with sketches and initial mockups.

Question	Answer
Are you a frequent user of WordNet Online application?	
If yes, what did you like about it?	
If yes, what do you think was missing?	
Are you a frequent user of other linguistic resources like MultiWordNet and EuroWordNet?	
If yes, what did you like about them?	
If yes, have you been using any other linguistic resources?	
Do you think there is a better way to design a multilingual linguistic resource?	

The following table has been used during the interviews to gather requirements for UKC interactive and collaborative platform, the table shows sample of the predefined questions. The interviews were conducted regularly to revise and update the requirements. Interviews were also accompanied with sketches and mockups.

Question	Answer
Did you use any platform for knowledge construction and maintenance other than UKC? If yes, was it a collaborative platform?	
If yes, what did you like about it?	
If yes, what do you think was missing?	
Do you know about web 2.0 collaborative and social features?	
Do you know about communities of practice?	
If yes, did you participate in any community of practice?	
If yes, which web 2.0 features may fit for an online community of practice for knowledge construction and maintenance?	
Do you think the proposed Workflows approach for UKC is an effective approach?	
If yes, how can we enhance it?	